A Novel Approach to Authorship Attribution

Niklas Zechner

"we present **a novel approach** for authorship attribution..."
Raghavan, Kovashka, and Mooney

"we propose **a novel approach** to mining
discriminative k-embedded-edge subtree patterns..."
Kim, Kim, et al.

"we are introducing **a novel approach** of authorship attribution..."
Iqbal, Farkhund, et al.

"We propose **a novel approach** to tackle the problem..."
Liu, Bing, et al.

"We have presented **a** simple but **novel approach**..."
Csurka, Gabriella, et al.

"we introduce **a novel approach** which uses
frequent item (term) sets for text clustering..."
Beil, Florian, et al.

"We propose **a novel approach** for categorizing
text documents based on the use of
a special kernel..."
Lodhi, Huma, et al.

"we discuss **a novel** mathematical **approach** to authorship attribution..."
Basile, Chiara, et al.

"we are proposing **a novel approach** of mining style variations..."
Hadjidj, Rachid, et al.

"we presented **a novel approach** for document clustering..."
Miao, Kešelj, and Milios

"we investigate **a novel approach** to spam filtering
based on adaptive statistical data..."
Bratko, Andrej, et al.

"This paper proposed **a novel approach**
for authorship attribution based on a
web-based self-training learning method..."
Guzmán-Cabrera, Rafael, et al.

Abstract: There has long been a need for more systematic work on the effects on authorship attribution from parameters such as amount of data and number of candidate authors. This study uses well known features — including frequencies of words and syntactic elements — to investigate the impact of varying such parameters. The same methods are also applied to some tests of topic dependence. The results show that small feature sets are sufficient even for large numbers of candidates, but that a large amount of data is needed regardless of those things. There are also several indications that features previously regarded as topic-independent, such as function words, may be highly topic-dependent after all, and that syntactic methods may be somewhat less so.

Previously published as a BSc thesis at the Department of Linguistics at Uppsala University.

1 Introduction

Despite over a century of research, the study of text classification is still chaotic. Many studies develop a "novel approach" to the problem, using new methods to try to outperform existing ones. Some try altering some of the parameters of the problem, to show that the success of their approach is independent of those parameters. But what it seems few have done is examine what does change with the various parameters, and try to analyse the performance of existing methods in a systematic manner. The purpose of this study is to do just that.

We look primarily at authorship attribution, and investigate how the accuracy of a few well known methods depends on a number of parameters, such as the size of the input data and the number of candidate authors. Are some methods better in some situations? Which parameters are the most relevant for successful classification? We also consider the issue of topic dependence interfering with authorship attribution. If an algorithm tells us that two texts are similar, how do we know whether it is because they are written by the same author or because they are on similar topics? Finally, we look briefly at why certain word features are more effective for classification.

In the short term perspective, knowing these effects is vital for evaluating methods, new and old. It can help us pick an appropriate method for a task, and estimate what sort of accuracy we can expect. In the long term, we need to strive towards a deeper understanding of what determines the results in classification, in order to improve development and application. This is why, rather than developing yet another new method and hoping to reach better results, we take the novel approach of a more theoretical, systematic study.

2 Overview

As early as the 1800s, scholars debated methods of analysing a text in order to determine who the author was. Since then, we have seen many new related problems, methods and applications. Today, automatic analysis by computers has given the field new momentum. Typically, we use a database of texts — a corpus — with known properties, and let the computer build a statistical model, which can be used to analyse an unknown text and guess its properties. As with some other computational linguistics problems — such as parsing — early versions of computerised classification often made use of linguistic knowledge [1], but today there is less emphasis on such expertise and more on statistics, which might contribute to the lack of understanding of what works and why.

Common problems of text classification include identifying the genre or topic of the text, sentiments such as whether a review expresses a positive or negative opinion, and properties of the author, such as gender, age, education level, or native language or dialect. In this study, we focus mainly on authorship attribution, also know as author identification. Even when we look only at authorship, there are actually quite a few variations of the problem. In the prototypical *supervised* classification problem, we have data for a number of known authors, and we want to identify which of them is the most likely to have written a given unknown text. In *unsupervised* classification, we have no known texts, so the classifier has to create the classes. A common case is that we have a number of texts and want to divide them into sets likely to be written by the same author. Another variant is the case where we have one known author and want to know whether a text is (or which texts are) written by that author. In some applications we want to go beyond yes/no questions, and find a similarity value, effectively a likelihood that a text is written by a given author, or that two texts are written by the same author. In this study, we calculate such a similarity measure, and apply it to a supervised classification problem; this is explained further in Section 3.

We use the same classification system to perform a number of different experiments. The system can be given different amounts of input data, and base the classification on different features of the text. Section 3 will explain how the process works, and the following sections will present the experiments.

In Section 4, we make use of a simple set of features, counting the frequencies of certain words in the selected texts. We vary three of the basic parameters of the test, to see how they affect the accuracy. First, the amount of data per author is varied. How much data do we need to get a reasonable accuracy? Is authorship attribution feasible for short texts like letters or even text messages, or do we need entire books worth of data? Second, the number of features is varied — in this case, the length of the list of words that we look for in the texts. Can we get arbitrarily high accuracy with enough features, or do we get to a point where more features are no longer useful? Third, the number of candidates is varied. Do we need more data, or more features, to get good results? Is authorship attribution feasible for large numbers of authors?

Previous studies have stressed the need for this kind of thorough analysis and deeper understanding [1, 2]. Most work has been done on small numbers of candidates, for which it is clear that accuracy decreases rapidly [3], but there have been indications that larger numbers of candidates might not be impossible to handle [4]. As for varying the amount of data, the combined wisdom of a few systematic studies [5] and a large number of studies of specific datasets suggest that 10 000 words per author is about as much as one needs. There does not seem to be any studies on how that limit changes with the number of candidates, nor on the effects of varying the size of the feature set.

In Section 5, we compare with other feature sets, including features based on syntax and character counts. Many studies on text classification have found simple methods like word count, using no linguistic analysis, to be the most effective [1]. We would like to try syntactic methods and see if they may be useful after all.

Could it be that simpler methods just fare better on small data sets, since they have more data to work with? Clearly, there are more characters than syntax trees in a sentence. Given enough data, presumably each method will level out at some level of accuracy — perhaps that level is sometimes higher for a more complex method, even if it started out lower? Perhaps some methods will better take advantage of larger numbers of features, or better handle larger numbers of candidates?

In Section 6, we use a corpus of novels to explore the problem of topic dependence — an important issue in authorship attribution. If an algorithm shows that an unknown text is similar to a known text, how do we know if the similarity is due to them being written by the same author, or due to similar topics,

genres, and so on? Knowing how to separate the influence of the author from that of the topic is important in order to avoid misinterpreting analyses. We also want to understand the underlying connections so that we can develop efficient methods for identifying an author, as well as for identifying the topic or other properties.

One hypothesis is that syntax, as well as some grammatical words, is reasonably independent of topic, at least if the context is otherwise the same — same genre, type of media, and so on. It has also been assumed that function words — words which carry little semantic information — are independent of topic. We will compare syntactic features to vocabulary, and see some indications as to whether this is true.

To achieve this, we think of the different books as different topics. On the one hand, we pick texts from different books and try to determine which ones are written by the same author, and on the other hand, we pick texts all by the same author and try to determine which ones are from the same book. By seeing how well each method works on the two tasks, we will get an idea of how much the classification depends on author and how much on topic.

In Section 7, we look into a seemingly minor detail in the implementation of the system. In order to test an algorithm for authorship attribution, we need more than one text that is known to be from the same author. Unless the corpus already provides that, we can solve it by dividing each text in parts. But there is more than one way to divide a text, and as we shall see, it can have a noticeable impact on the accuracy reported by the test. Other studies [5] have found similar differences, without elaborating on the implications.

In part, this is a straightforward test of evaluation methods — if different ways of testing give very different results, it may be difficult to compare the reported accuracies of different studies and systems. But as we shall see, the difference can also tell us something about topic dependence and the reliability of a method when used for authorship attribution.

In Section 8, we try to get a deeper understanding of why some features — specifically, some word counts — are more effective than others. Are more common words better, or are there some words that are so common that they are useless for authorship attribution, since everyone uses them? Can we calculate in advance which words are going to be the most useful?

3 Method

In order to make a successful statistical model, we need good statistical data. We will compare data from three different corpuses; we refer to them as Boards, Blogs and Novels. The first contains data from the Irish web forum boards.ie [6], where we take all posts from 2006 and 2007 by authors with at least 60 posts, which makes 5450 authors. The second comes from a set of blogs gathered from blogger.com in 2004. It contains 19 320 blogs by as many different authors [7]. These two corpuses are quite similar in that they come from modern web-based sources, and contain many authors, but not very much text from all of them. The third corpus is different. It consists of 290 novels, written between 1881 and 1922, by 25 different authors [8]. Having several separate large texts by the same author is very valuable for learning about authorship attribution. We hope that it can give us some idea as to which features really are specific for an author, and which vary depending on context. We will essentially consider the different books by each author as "topics". This is not ideal — some books may of course be on the same topic, and they are in the same medium and style — but large corpuses marked for topic as well as author are not easy to come by, particularly since topic is much more vague than author, so we have to make do with this. To our advantage, separate books are at least much more well-defined than topics.

With the web-based corpuses in particular, it is possible to consider various paratextual data, such as posting time and markup, but we exclude those things and extract only the text itself from each corpus. We also ignore the information that could be extracted from looking at the posts separately. It is likely that average length of posts would have been a useful feature, and there are algorithms that count distributions of feature values over posts rather than looking at a single feature value for the entire text, but we will use neither in this study. In order to simulate unknown identities, we divide each candidate's texts in two halves, a and b. The algorithm then compares each a-part to each b-part, calculating a similarity rating for each pair. For each a-part, it tells us which b-part has the highest similarity. If it is the one which is actually by the same candidate, that is considered a successful match. The percentage of successful matches is considered the accuracy. This way, we have a method which can answer questions like "which of these authors wrote this text", but since we have similarity measures as an intermediate step, we can also look at what might happen if we ask "how likely are these two texts to be written by the same author".

When we have our suitably divided data, the next step is to choose which features of the text to use for the statistical model. All the features used in this study are based on counting the relative frequencies of certain things in the texts. The first feature set is a count of words. For each of the chosen

words, the program goes through each text, counting how many times that word occurs. This is divided by the total number of words in the text, to get the relative frequency. As words, we count not only typical words but also punctuation marks. Capitalisation is disregarded.

For the most part, we will use the most common words in the corpus, counted for each corpus separately. The table shows the top ten words for each corpus.

	Boards	Blogs	Novels
1	.	.	,
2	the	,	the
3	,	i	.
4	to	the	”
5	a	to	and
6	i	and	of
7	and	’	to
8	of	a	-
9	in	of	a
10	it	it	i

Table: The ten most common words in each corpus.

As we can see, there are many similarities, but a few differences, most of which are not surprising. In Blogs, we have a high number of “I”, as expected, since blogs often contain texts about the author, like a diary. It is also worth noting that these blogs are from a time when the concept of blogs was quite new; perhaps today they have branched out to other subjects with fewer “I”. In Novels, we see more commas than full stops. It is as expected that these more formal texts have longer sentences, and the writing style of the time period may also contribute to the greater number of commas. The same things explain the lack of apostrophes, and more quotation marks are also expected in a novel. In Boards, we might note that “a” is more prominent than in the others; perhaps in a setting where many people write together, more new entities are introduced, which could increase the frequency of the indefinite forms.

For another feature set, we extract syntactic patterns. We use the well-known Stanford parser [9] to perform a dependency analysis, and extract the syntactic relations. We will look at two different variants of this method. For an example, consider the sentence “He went to London”. With the first method, we extract only the relations, so in this case we get

1x subject (“he” is the subject of “went”)
1x sentence root (“went”)
1x preposition “to” (“London” is the prepositional object of “went”)

We are using the collapsed version of the parsing scheme, which treats certain words — including prepositions and conjunctions — as marking relations, rather than participating in them, which is why we only get three relations from these four words. This, along with the fact that the parser sometimes fails to process a sentence, means that we will get slightly less data than for the word counts.

The other variant counts not just the relation itself, but also the types of words involved. So in this case we would get

1x subject relation between a pronoun and a past tense verb
1x sentence root with a past tense verb
1x preposition "to" with a past tense verb and a proper noun

We will also look briefly at counting characters. There have been studies using both letters and punctuation as features [4]; we will simply count all characters, and use the most common, just as for words. The word features are counted independent of case, but the character features are case-dependent. Even so, there are a limited number of reasonably common characters, so this feature type may have limited usefulness. Other studies have also extended the idea to n-grams (that is, sequences of characters) [1], but we will not go into that here.

For the mathematical evaluation, we use cosine similarity on standardised values. We start by extracting the relative frequencies of each feature for each text from the corpus. We would like all the feature values to be treated similarly, regardless of how common the feature is, so we adjust the distributions of each feature so that the average is 0 and the variance is 1. This gives us a sequence of values for each text. To compare two texts A and B, we calculate the cosine similarity,

$$\text{similarity} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \ .$$

This classifier has proven reasonably fast and effective, and the standardisation greatly improves the result — others have come to similar conclusions [4]. Aside from that, experimenting with classifiers is beyond the scope of this study.

4 Exploring parameters

Many studies on classification present the results of their novel approach as a percentage — our method correctly identified this many percent when working on this data, and so on — so let us begin by doing the same thing. We work on data from Boards, picking the first 100 candidates by user ID (so essentially a random sample). They have written 13243 words on average. Using only the ten most common words, our method has an accuracy of 52%.

So what does this tell us? That this system is not as good as one that gives 60% accuracy? That an application of the system is likely to give an accuracy around 52% on similar data? As it turns out, this is not a very good analysis. Some of the authors have written a lot more than others, and this is quite relevant. Let us look at how much each of them has written.

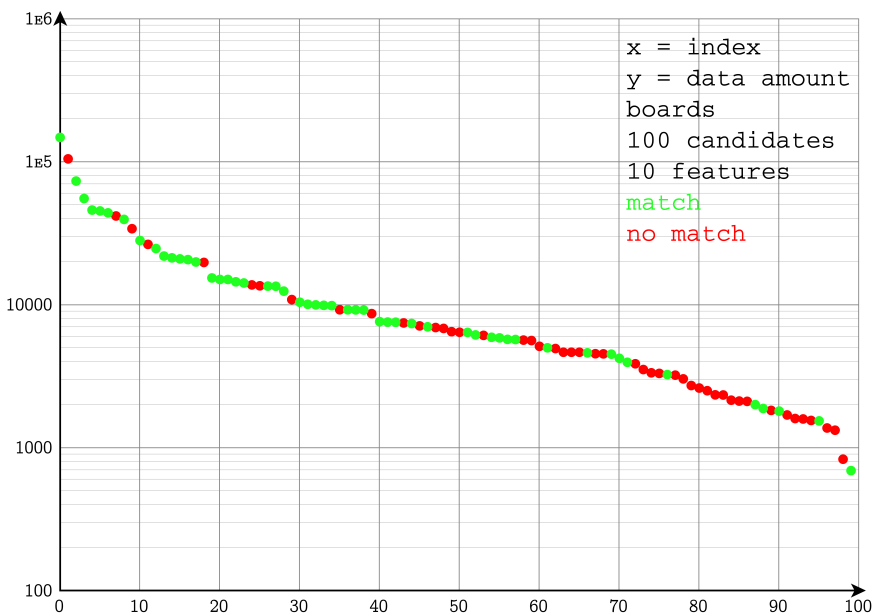

Figure 1: Distribution of data amount for the first 100 authors in Boards.

In Figure 1, each point represents an author, and the position on the y-axis is the data amount — that is, how many words we have for each author. The points are also ordered by decreasing data amount. Note that the y axis is logarithmic; that is, each major step on the y-axis represents a tenfold increase in data amount. Green points represent authors who were successfully identified, and red dots represent those who were not successfully identified. We can

clearly see that the accuracy is very different depending on how much data we have. In fact, although the average data amount is well over 10 000 words, if we tested the system on an author with 10 000 words, we should really expect an accuracy significantly higher than 52%.

The chance of identifying a given author clearly depends on how much data we have for that author, and (at least for a symmetric system like this) it must also depend on how much data we have for the other candidates. So what we should do instead is extract the same amount of data from each author. Then we can report what accuracy we get for that amount of data, or better yet, we can vary the amount of data and report the different results.

Is it a problem that we assume the same amount of data for all authors? Certainly in an application we would not have that, but it is as good an assumption as anything else — better to choose a known distribution than to report the results of a random distribution.

In order to vary the amount of data, we would first like to know how much data there is, so we look at the distribution of data in Boards and Blogs as a whole.

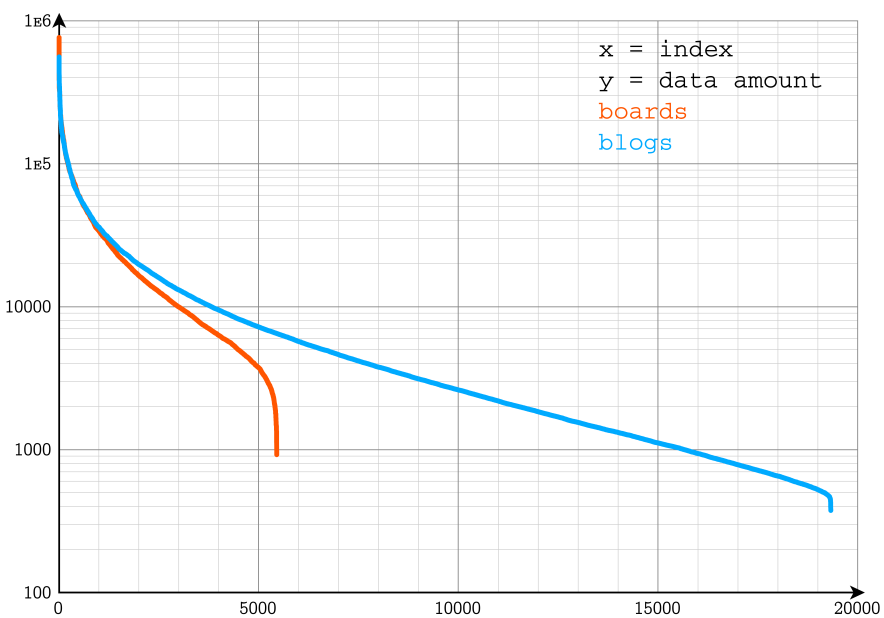

Figure 2: Distribution of data amount for all authors in Boards and Blogs.

This not only gives us an idea of the amount of data we are looking at; it also tells us how far we can go in comparing accuracy at different amounts of data. If we want to, for example, have test sets with 5000 words per author, then we will need 10 000 words per author in total — since we need two parts to try to match — so the graph tells us that we must not pick more than about 3000 authors for Boards, or slightly more for Blogs. Conversely, if we want to look at 5000 candidates, the graph tells us that we can go as high as about 7000 words per author for Blogs, but less than 4000 for Boards, so half of that for each part. We want to go as high as possible in amount of data, while still having enough candidates to get a useful number on accuracy; we will will start by looking at 100 candidates, for Boards. To begin with, we are only looking at the word frequency method.

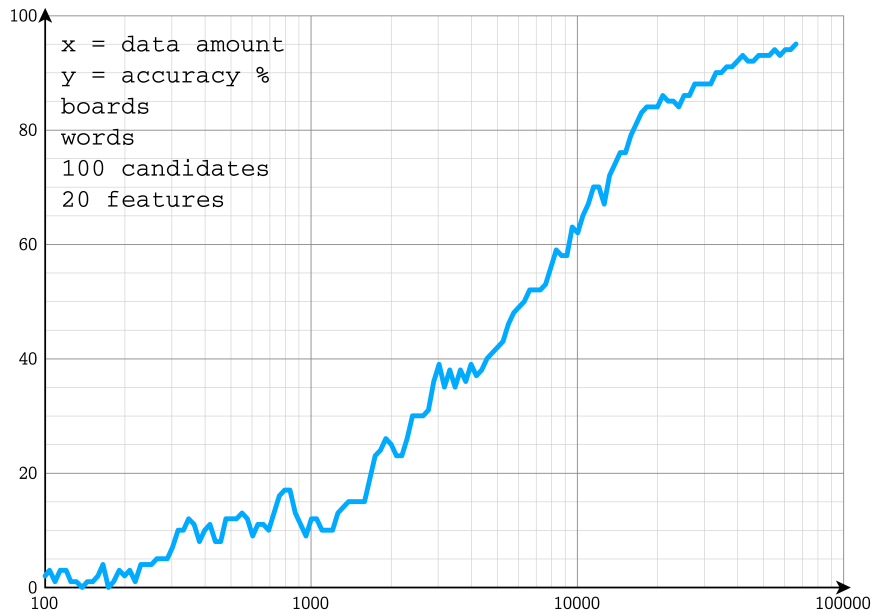

Figure 3: Accuracy as a function of amount of data, for Boards.

Figure 3 shows the results of trying to identify an author out of 100 by counting the frequencies of the 20 most common words. The amount of data is counted per half, so where it says 10 000 we can think of it as having 10 000 words per "known" author, as well as 10 000 words for the "unknown" author. Or if we want to use the system to see if two texts are likely to be written by the same author, we would then have 10 000 words for each of the two texts. It is difficult to see in Figure 2, but the 100 most prolific authors each have at least 140 000 words, so we can use up to 70 000 for each half.

The graph gives us a rough idea of how much data we can expect to need to make authorship attribution viable, at least for this number of authors and features: If we want more than a random guess, we probably need a few thousand words per author; if we want to get it right more often than not, we should have at least 10 000; and if we want a system that can reasonably be called reliable, we would need to get close to the 100 000s. If we had more data, would the accuracy continue rising, getting arbitrarily close to 100%, or would it level out at some point? Hopefully we will see some clues to answering that later on. This data is for the Boards corpus, so let us compare with the same graph for Blogs.

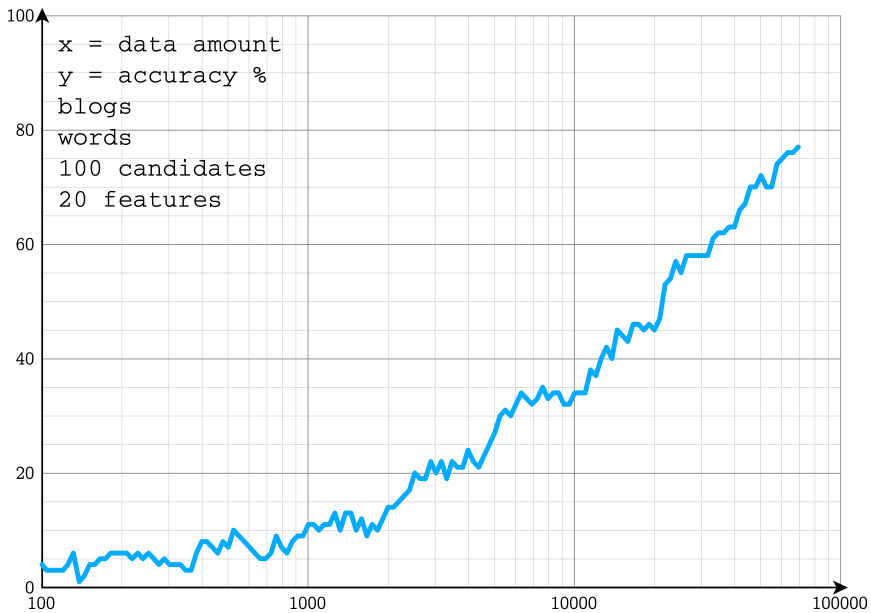

Figure 4: Accuracy as a function of amount of data, for Blogs.

The pattern is similar, with noticeably lower accuracies than Boards — or perhaps rather, we need a bit more data to reach the same results as for Boards. We can only speculate as to why that is — should not blog texts be more personal and therefore easier to identify? We will get back to that question in Section 6. So far we have looked at 20 features, but in a real-life application, we have little reason to limit the system to such a small number of features. Some classifier algorithms get very slow when using many features, but simple ones like this one can easily handle quite a few more. Let us see what the same graphs look like with higher numbers of features.

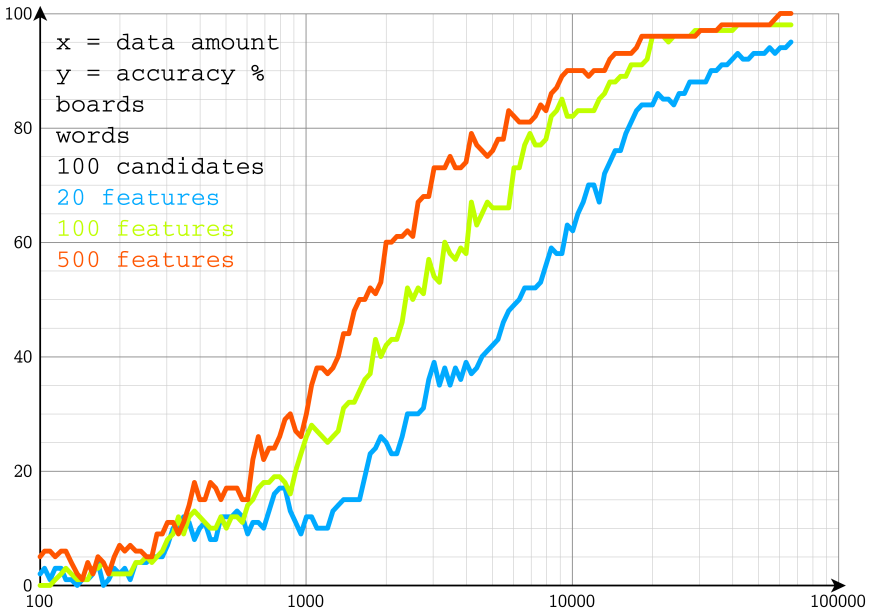

Figure 5: Results for Boards with 20, 100 and 500 features.

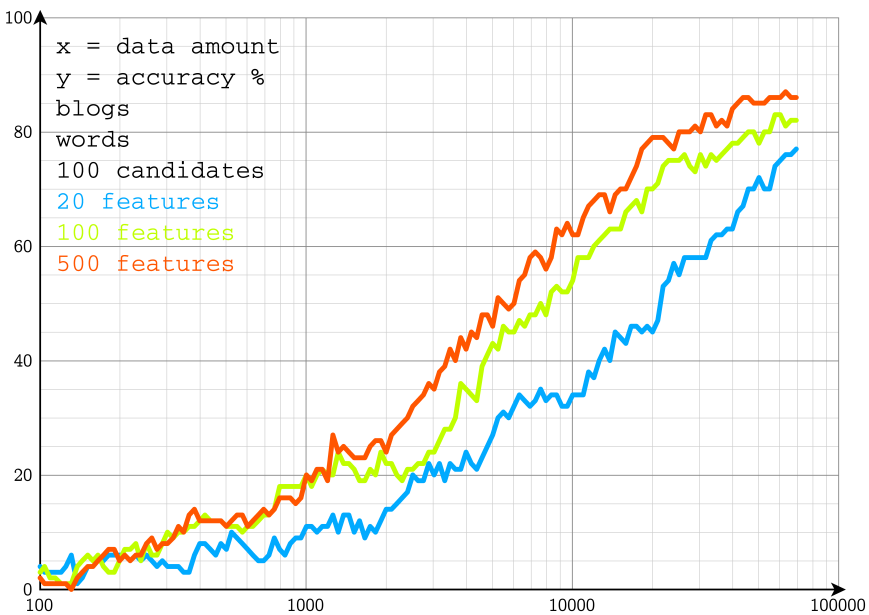

Figure 6: Results for Blogs with 20, 100 and 500 features.

As expected, we get higher accuracy when we use more features. The difference is not overwhelming — we still need tens of thousands of words to get somewhat reliable results. At least for Boards, it looks like it is possible to get very close to 100%.

What happens with more features, or fewer? We can look at a different kind of graph, where we vary the number of features on the x axis, for a fixed amount of data.

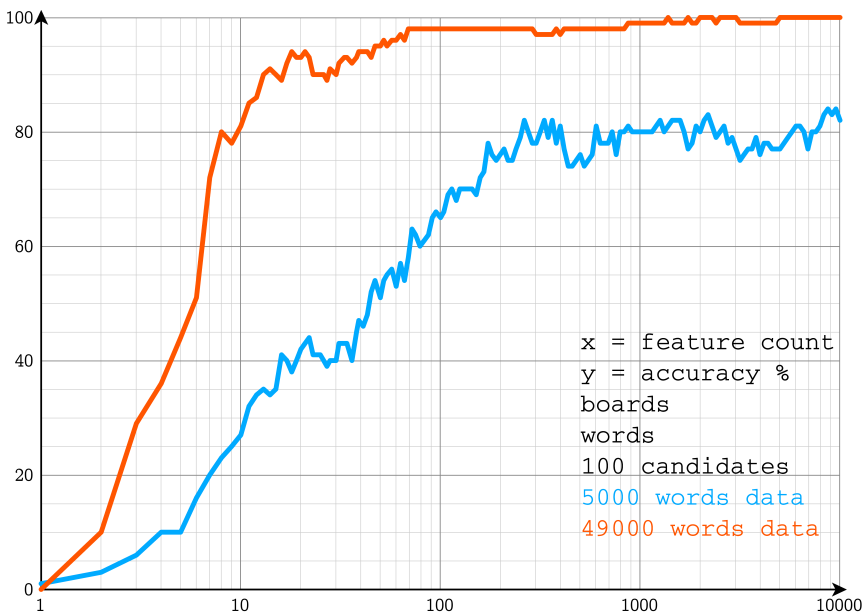

Figure 7: Varying the number of features, for Boards.

As Figure 7 shows, with a decent amount of data, it takes a remarkably small number of features to have a good chance of identifying the author. On the other hand, the lower curve seems to have levelled out at about 80%, which suggests that if we have a bit less data, no amount of features is going to be enough to get a really reliable identification test.

The higher curve is using 49 000 words per half, so 98 000 words in total per author, which is slightly more than the length of *The Hobbit* [10]. So if we were hoping to identify a person based on a brief email, we would be disappointed.

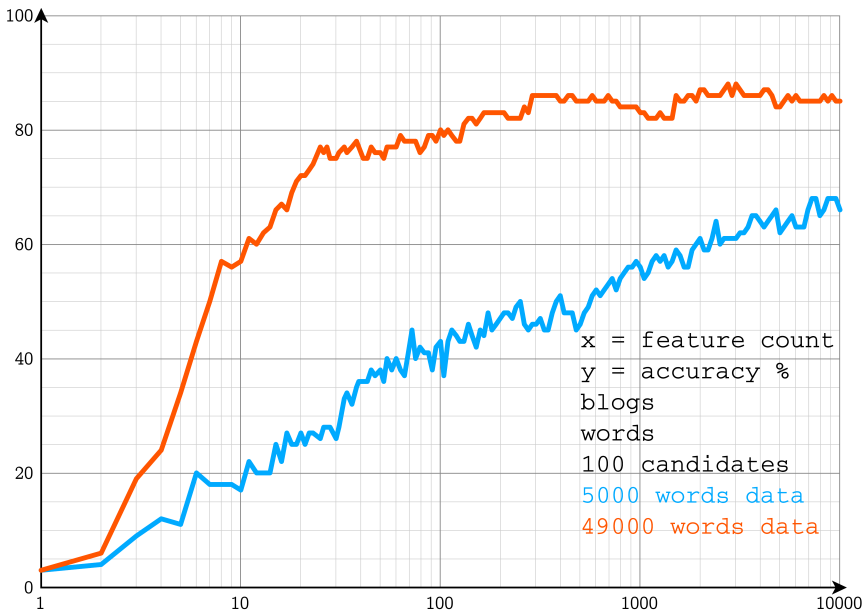

Figure 8: Varying the number of features, for Blogs.

Figure 8 shows the results for Blogs. Looking at the upper curve, it still levels out shortly after ten features, but at a lower level than for Boards. The lower curve seems to be steadily rising even at 10 000 words.

So far we have only looked at 100 candidates. There is good reason for this — the more candidates we add, the less data there is available in the corpuses, so we will not be able to see what happens for large amounts of data. Still, let us look at how the graphs differ with more candidates.

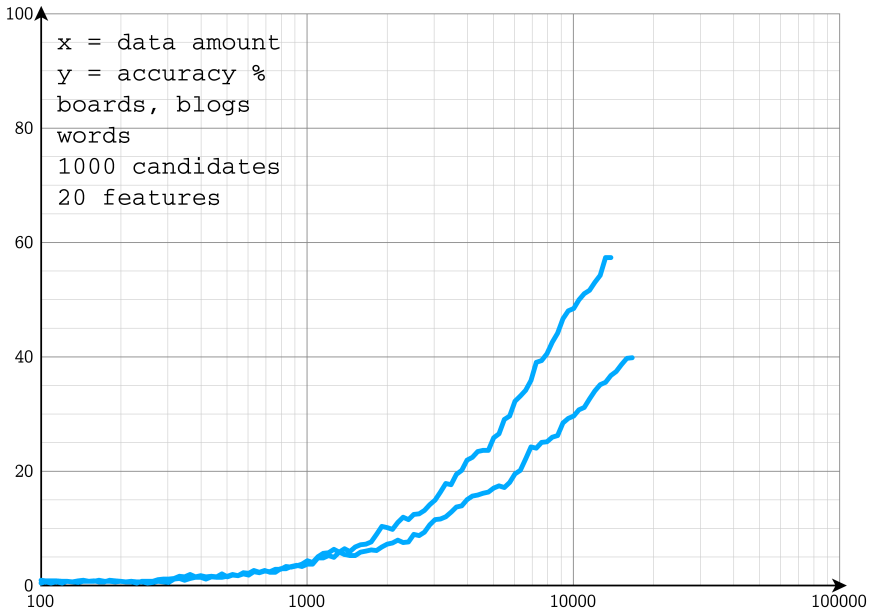

Figure 9: Varying the amount of data, for 1000 candidates.

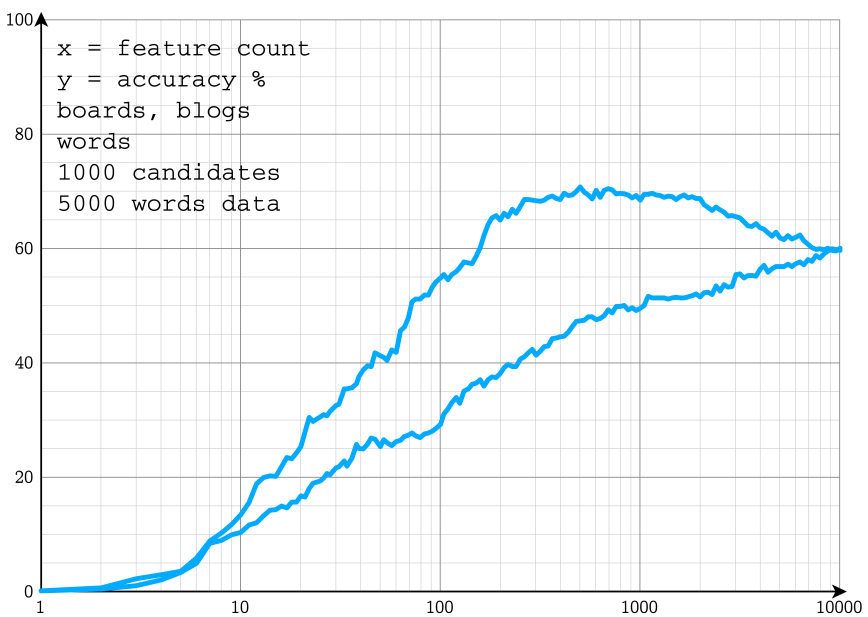

Figure 10: Varying the number of features, for 1000 candidates.

Figure 9 shows the same things as Figures 3 and 4, but for 1000 candidates. The higher curve is for Boards. Comparing to the earlier graphs, we see that the accuracy is lower, but not very much. For example, for Boards, at a data amount of 10 000 words, the accuracy is now around 50%, whereas with 100 candidates it was around 60%.

Figure 10 shows the equivalent of Figures 7 and 8, for 1000 candidates. These graphs are also remarkably similar — despite having ten times as many candidates, the accuracy is only slightly lower. At 10 000 features, Blogs reaches 60%, as opposed to around 65–70% before.

Finally, we can also look at a graph where the number of candidates is the variable on the x axis.

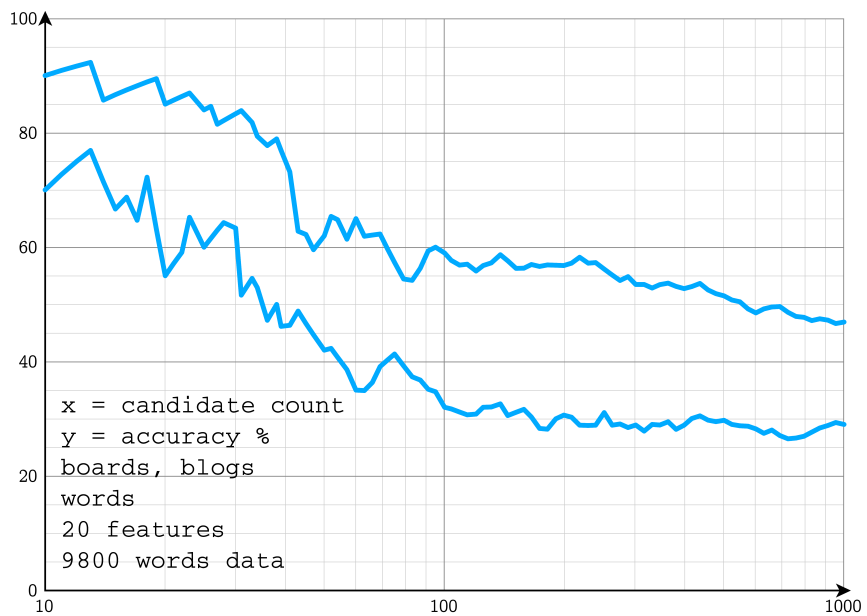

Figure 11: Varying the number of candidates.

Figure 11 verifies that after about 100 candidates, the difference is smaller than one might have expected. The methods may not be very effective for small amounts of data, but they are all the more effective for large numbers of candidates.

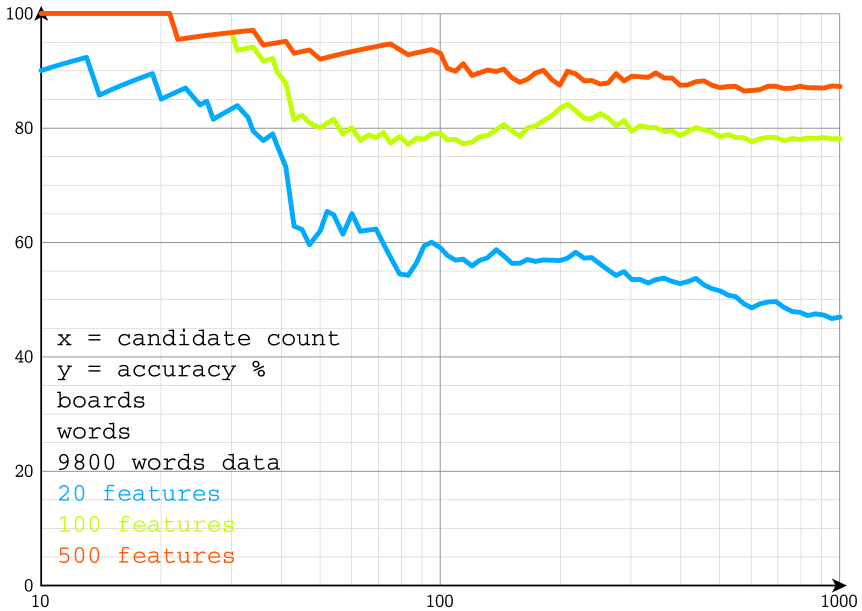

Figure 12: Varying the number of candidates. 20/100/500 features, for Boards.

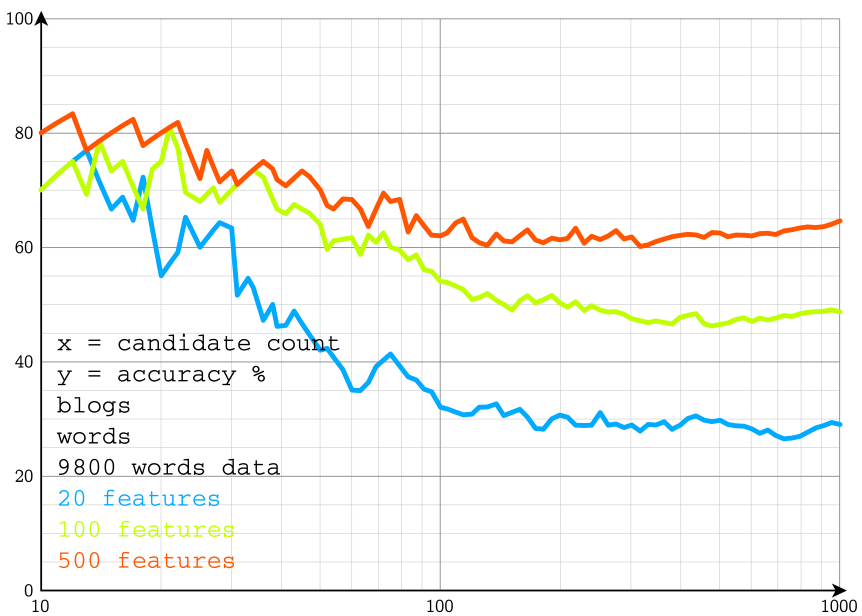

Figure 13: Varying the number of candidates. 20/100/500 features, for Blogs.

With more features, the difference is even less pronounced. At least in Figure 13, we can even see a hint of an increase at the end. We can only assume that this is due to the increase in data — we are keeping the amount of data per candidate constant, so the total amount of data available increases.

It would have been interesting to see what happens with a larger number of data, if the accuracy of 95% or more could be maintained, but unfortunately neither of the corpuses have 1000 authors with that much data. What we can do is increase the number of features. Still, the results we have suggest that it would be possible to get good results even for large numbers of authors — assuming we can find at least a novel's worth of data for each candidate.

5 Exploring features

In the previous section, we saw how the accuracy of the word frequency method changes with the various parameters. Now we will compare with a few other methods, to see if they all behave the same way, or if some methods are more useful in some situations.

As explained in Section 3, the "words" we are using include not only actual words, but also punctuation. This is quite reasonable in an applied situation, but nonetheless we may be curious as to what would happen if we only use actual words.

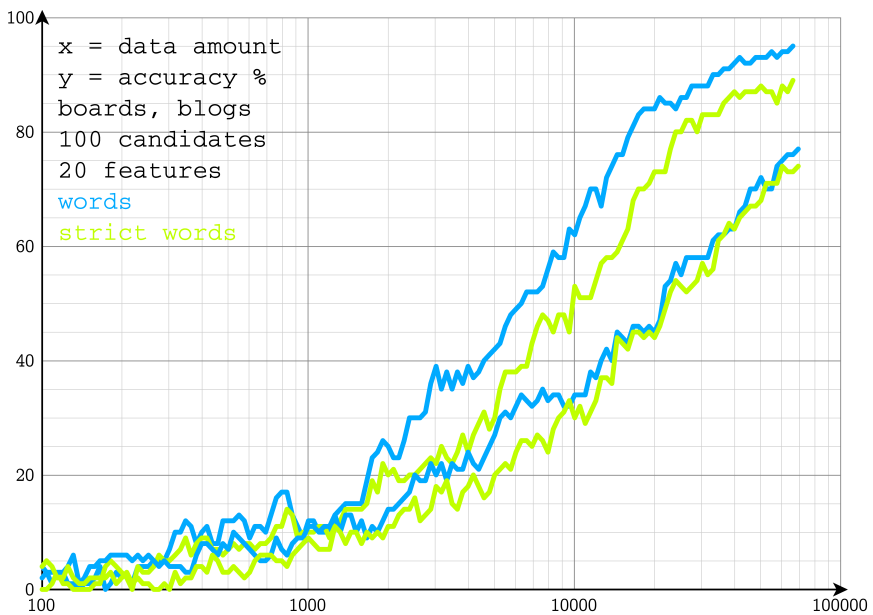

Figure 14: Comparison of the standard words and strict words.

Figure 14 shows the curves from Figures 3 and 4, along with the corresponding curves for when we only use actual words. The differences are about as can be expected; the choice of slightly less common features brings the accuracy down a little, but the results are about the same. Note that the total word count — the numbers on the x-axis — is still in terms of the wider definition. We will focus on the wider definition of words from now on.

Another even simpler type of feature is counting characters. Let us look at how that method compares to using words.

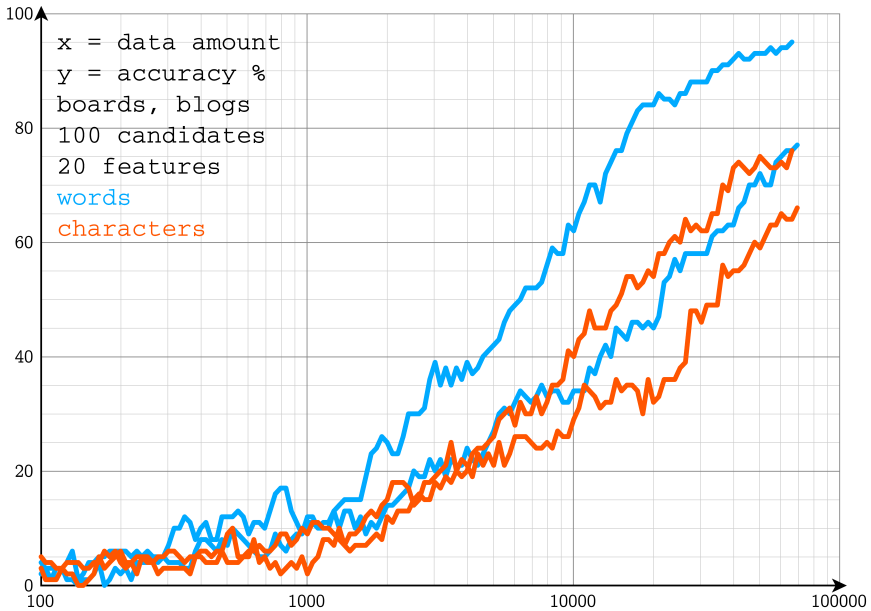

Figure 15: Comparison of words and characters.

Figure 15 shows the corresponding results for characters. Note that we are still counting amount of data in terms of number of words, not number of characters. We see that characters are a surprisingly viable method; it is apparently entirely possible to identify an author solely on the basis of how many times they use each letter. This is very interesting on a theoretical level — we would not typically describe an author's style as "using a lot of G" — but the results are not as good as for words, even when we only look at 20 features. The characters method does distinguish case, and includes punctuation and other symbols, but even so, it does not seem meaningful to extend the characters method to hundreds of features, so we will not look any further at this method. An obvious extension would be to use n-grams, that is, sequences of characters, but we leave that for future work.

This brings us to the most interesting feature types, using syntax. As explained in Section 3, we have a simple and a more complex syntactic method. We compare both of them to the words method.

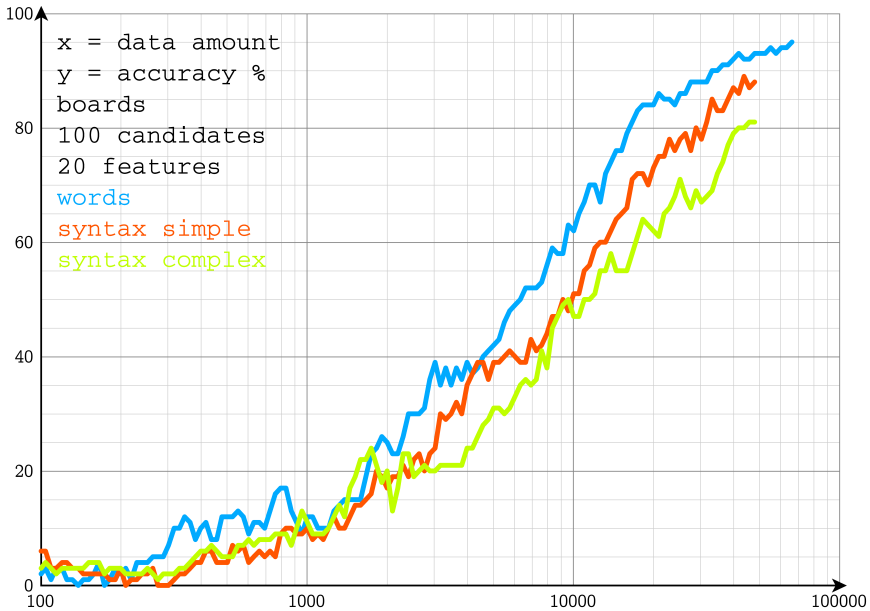

Figure 16: Comparison of words and syntax, for Boards.

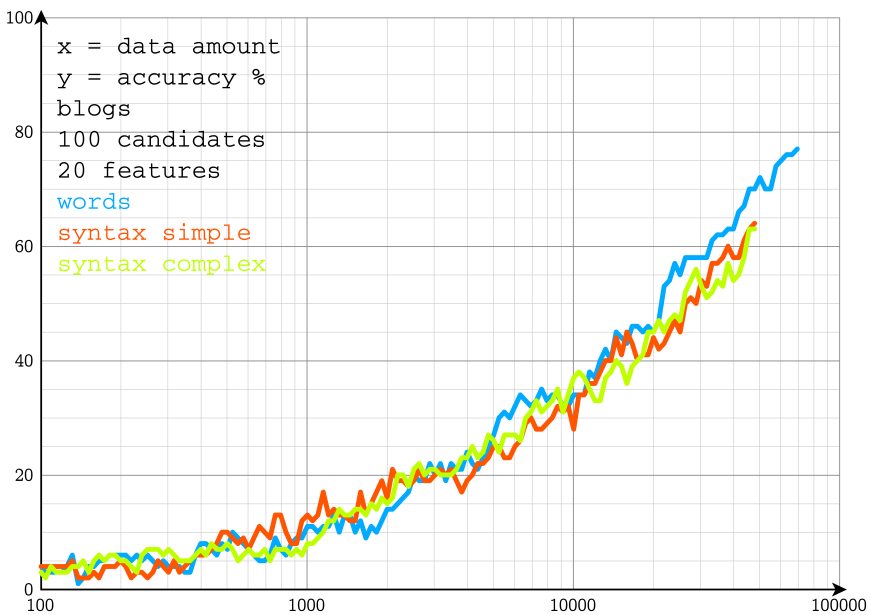

Figure 17: Comparison of words and syntax, for Blogs.

For Boards, the syntactic methods are a bit behind, but not by much, and for Blogs, they are both at more or less the same level as words. But perhaps we should not place to much importance on those differences. As is probably becoming apparent, there are many things that can influence the accuracy, so a shift in a few percentage points is not in itself particularly noteworthy. Notice that the curves for syntax do not extend quite as far to the right as those for words, since we have slightly less data for syntax, as explained in Section 3.

More interesting is that all the curves have essentially the same shape. We might have expected that some methods would be more useful for smaller amounts of data; for example, we might expect characters to be more useful initially, but level out sooner, since there are more characters than words. Similarly, one could imagine that syntax would start out slow and catch up when we have more data, due to being a more complex method. But as we can see, the curves all move in parallel when we vary the amount of data.

What about when we vary the number of features? The simpler syntax method has to some extent the same problem as characters; the parser does produce an open set of relations, but most of them are likely to be very rare (and, judging from a brief look at the output data, very wrong), so we would expect this method to fall behind quickly after we pass the reasonably common relations.

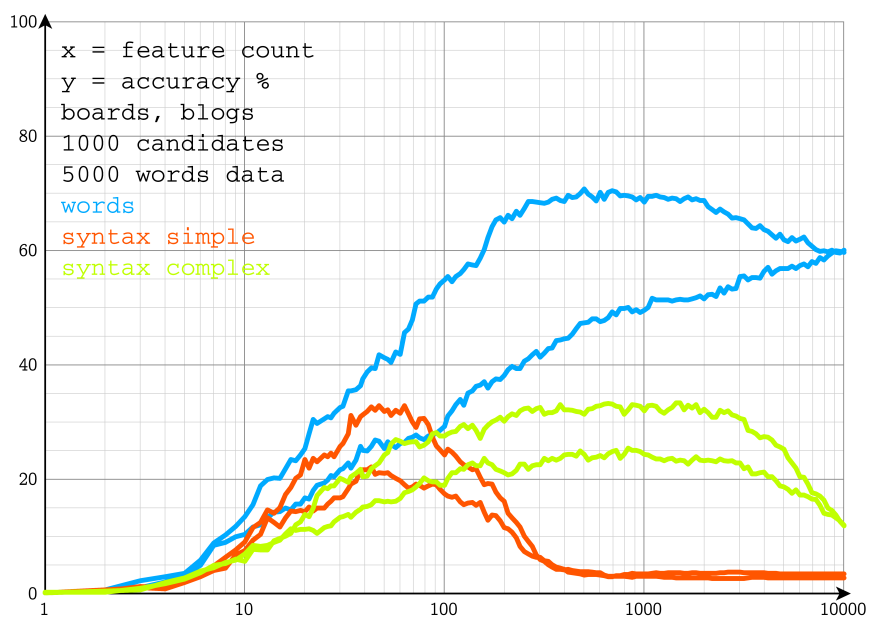

Figure 18: Comparison of words and syntax, varying the number of features.

Indeed we see that after only 40 or so features, there is nothing more to gain from adding more features to the simple syntax method, and the accuracy soon drops. If we were to look at the graph for 500 features, we would see the complex syntax method far above the simple one, but that would be deceptive — as we can see here, the complex method never actually surpasses the simple one, so the more complex analysis as well as the use of more features are a waste of time.

The curves for words also seem to reach a peak after a while — at least the one for Boards, and presumably the Blogs curve would do the same eventually. Why and when that happens might be an interesting topic in itself, but for practical purposes, we will probably not want thousands of features anyway, as we shall see in Section 6.

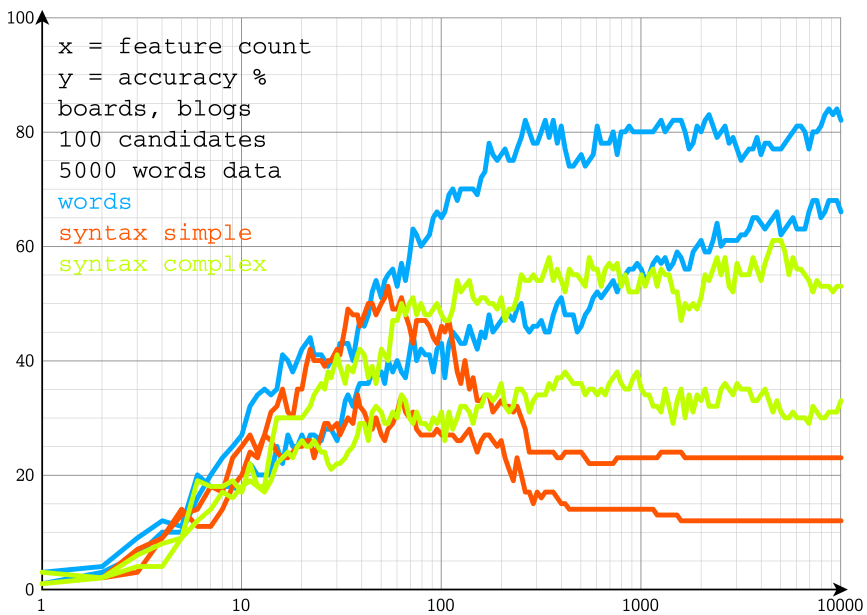

Figure 19: Comparison of words and syntax, varying the number of features, for 100 candidates.

Figure 19 shows the same thing for only 100 candidates. Here, it seems that the complex syntax method eventually goes higher than the simple, but only by a small amount and after many more features. We will focus on the simpler variant from now on. But for completeness, let us also look at the dependence on number of candidates.

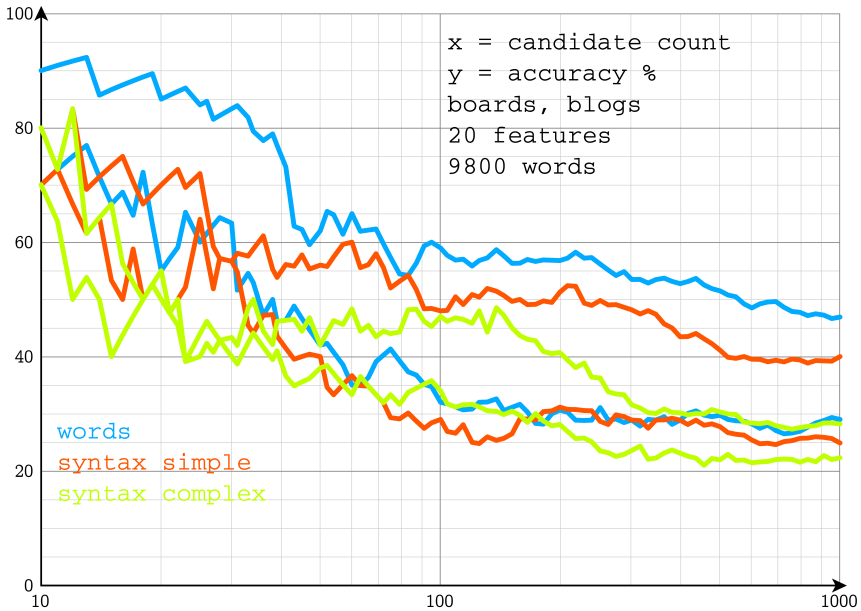

Figure 20: Comparison of words and syntax, varying the number of candidates.

As before, the curves are quite flat in the 100–1000 range. There is no remarkable difference in the behaviour of the different feature types.

6 Author and topic

In this test setting, several methods have proven viable. But in a real-life setting, there is a risk that we have not yet taken into account: The features might depend on topic as well as author. If the authors in the test set write about different topics, we might get greatly overoptimistic test results. When we later try to identify texts by an author on a different topic than in the known texts, the system is likely to fail, perhaps finding greater similarities with other texts on a more similar topic.

We could argue that some features are likely to be more topic-dependent than others. Certainly if a writer uses the word “football” a lot, that could give us high results on the tests, but would not be a reliable indicator in most applications. One idea is to use only function words, but there is no clear definition of what is a function word. Looking at the frequency list, it seems that the most common words could all be considered function words, so there

is no need to look specifically at function words as a separate feature set. But this does mean that using thousands of word features is a very dubious choice.

But are any other features better? If simple function words, syntax patterns, and even character frequencies depend heavily on the author, could they not just as easily depend on topic? It may seem unlikely that the number of "the" or "and" should depend very much on topic, but on the other hand, there is little reason to assume that it would differ between authors either.

To really answer that, we would need a corpus where each text is marked with topic as well as author. This is clearly problematic — not only are few texts organised by topic, but topic is a much less well-defined categorisation than author. We also need to have large amount of texts; tens of thousands of words, on a well defined topic, by ideally hundreds of different authors. But there may be a way around the problem, to get at least some sort of approximation for topic. In this section, we will use the Novels corpus mentioned in the introduction, and let the different books act as "topics". Most of the novels contain enough data on their own to make classification feasible, according to the numbers we have just seen. So on the one hand, we can attempt to identify the author of some text from one novel by comparing only with text from another novel. That way, we have hopefully isolated only author differences, not topic differences. On the other hand, we can pick several books all by the same author, and attempt to identify whether two texts are from the same book, thus isolating topic differences instead. If we can see that some methods work better in the first case, it strongly suggests that that method is more author-dependent than topic-dependent.

We divide the Novels corpus in a few different ways. First, we combine all the books by the same author chronologically into one big text, so that we are effectively identifying the author of one set of books by comparing with other books, thus hopefully avoiding most of the topic influence. This is quite similar to what we have done with Boards and Blogs. Second, we look at only books by one author, and try to identify the book, thus avoiding differences that come from different authors. For most of the authors in the corpus, there are not enough books that we can make a meaningful test, so we pick only the most prolific writer, Henry Rider Haggard. Third, we take one book by each author, so that we are effectively comparing both topic and author at the same time. We choose the longest book by each author, to try to extend the test as far as possible in terms of amount of data. We call these sub-corpuses Bundled, HRH and Single.

Let us begin by looking at the distributions of data in these, as compared to Boards and Blogs.

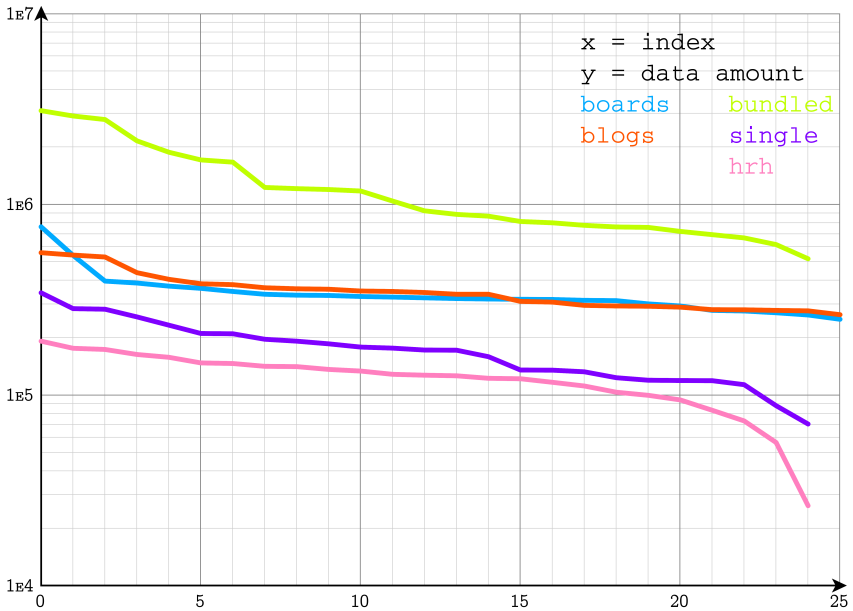

Figure 21: Distribution of data for the Novels sub-corpuses, along with the most prolific authors in Boards and Blogs.

Figure 21 contains the same curves as in Figure 2, but now for a much smaller set of authors, and the corresponding curves for Novels. We see that they are comparable in size; the most prolific writers on both Boards and Blogs have written the equivalent of a couple of novels.

We have 25 potential candidates for both HRH and Single, but the last few have so much less data that we are going to leave them out and make do with only 20 candidates. Before we get to Novels, let us see what the results look like for Boards and Blogs when we have only 20 candidates.

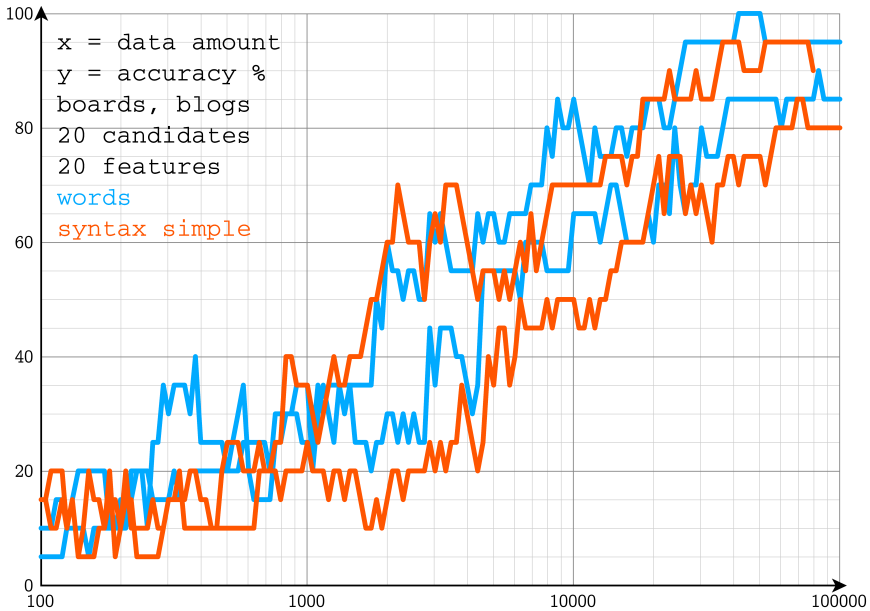

Figure 22: Results for Boards and Blogs, with 20 candidates.

For so few authors, 20 features are enough to get good accuracy. That also means that the simple syntax method is better than the complex, so we will focus on that.

Note that since we now have only 20 candidates, a single candidate is 5%. This explains why these graphs are more noisy than the earlier ones. It also means that the expected accuracy of a random guess would be 5%, so results below some 10% or so are just too low to measure this way.

Now we shall see how the same test works on Novels.

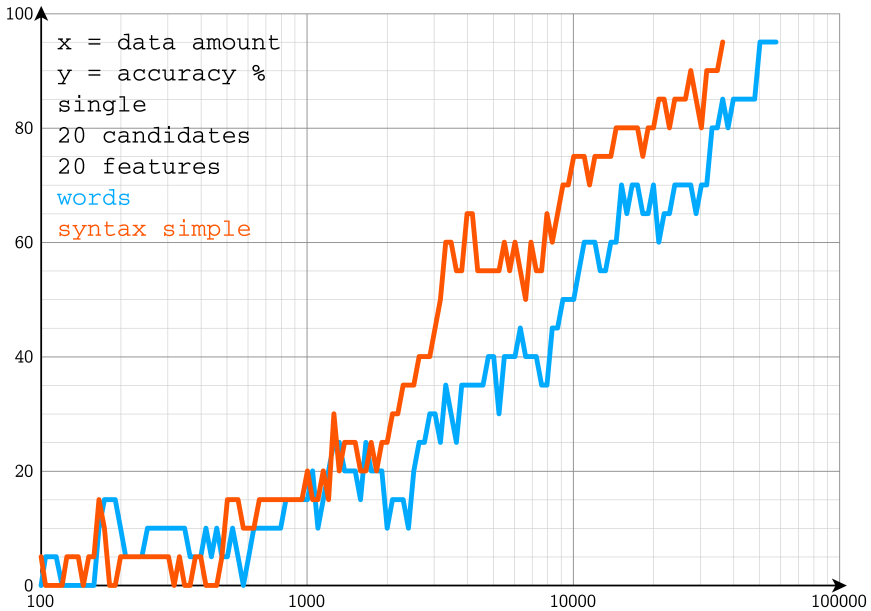

Figure 23: Results for Single.

As mentioned, for Single, the algorithms is effectively classifying topic along with author, which should make things comparatively easy. We can probably think of this sub-corpus as the closest equivalent of Boards and Blogs. Writers on a forum or a blog may write about a wider range of subjects than one would in a single novel, but the subjects might not change very much over time, so we can expect the two halves of each dataset to contain about the same mix of topics.

For words, the result is indeed comparable to Boards and Blogs, but slightly lower. That brings us back to the question of what causes the difference in accuracy between the corpuses. In novels — and even more so in a hundred year old novels — the style is more formal, so it makes sense that it would be less individual. That would also explain why Blogs gets a lower accuracy than Boards; even though a blog post is often very personal and informal, it is still a more lasting text than a forum post, and the author is likely to take more time to ensure correct spelling, grammar, and so on. Why the results for syntax are clearly higher for Single is less obvious; perhaps the longer sentences of a novel makes for more easily classified syntactic data. This is all rather speculative, and it may be that we should not read too much into it.

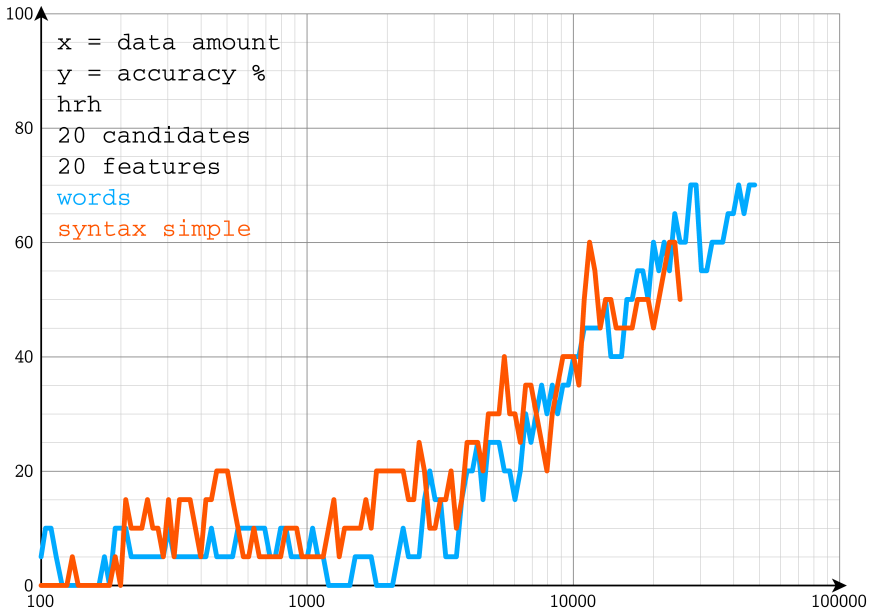

Figure 24: Results for HRH.

For HRH, we get noticeably lower results, which is entirely expected; books by the same author should reasonably be more similar, and thus more difficult to identify, than books by different authors, otherwise the task of authorship attribution is more hopeless than we thought. We also see that syntax is now as low as words.

But perhaps the most important thing in this graph is the fact that the curves are not even lower. Despite using only 20 features, both curves are surpassing 50%. All of the 20 words would clearly be considered function words, which means that neither function words nor syntax can be called topic-independent. This cast serious doubt on any study claiming reliable authorship attribution based on those methods, and we have little reason to think that characters or n-grams would be any better.

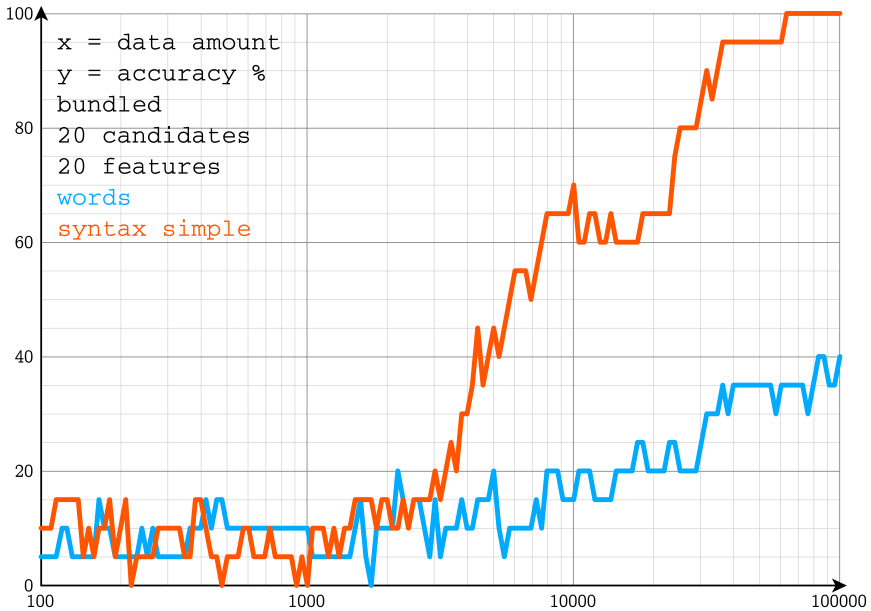

Figure 25: Results for Bundled.

The results for Bundled are all the more surprising. Syntax is now far ahead of words, and at least a little bit higher than for HRH. This suggests that although not completely topic-independent, syntax is at least much less topic-dependent than words.

In all the Novels graphs, we see that until around 1000 words, the accuracy is hovering around 5%. As mentioned above, this means that a test like this gives no meaningful data for those amounts of data. Looking at Figure 25, we might be led to believe that the methods are equally good until around 3000 words, but in fact they are just both too low to register.

There is a different way we can visualise how the methods identify topic and author. As mentioned in Section 3, this classifier not only tells us which is the most likely candidate, it tells us the similarity with each candidate. So for each a-part, we can measure the similarity for each b-part — the one from the same book, the few from other books by the same author, and the many from books by other authors. By comparing the similarity scores, we can see how big the difference is between same-book pairs, same-author pairs, and different-author pairs. Rather than just an average similarity, we will look at the entire distribution of similarities.

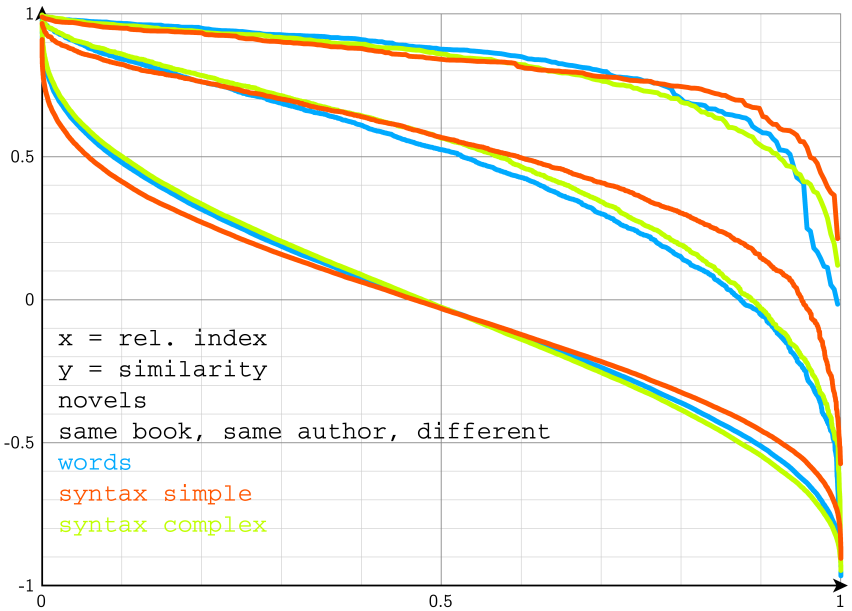

Figure 26: Distributions of similarities.

The somewhat complicated graph in Figure 26 contains curves for each of the three methods — words, simple syntax and complex syntax — and for each of the types of pairs — same book, different books by the same author, and books by different authors. The distributions have been adjusted so that they all go from zero to one. The top curves are those from the same book, which naturally have the highest similarity, and the second set of curves are those for books by the same author. Since the different-author pairs are the vast majority, it is no surprise that the median of those curves is close to zero.

For an example, we can start by looking at words, and which pairs have a similarity higher than 0.5. For the different-author pairs, it is just below 10%, for the different-book pairs, it is just over 50%, and for the same-book pairs, it is over 90%. We can use this as an indication of how likely it is that two texts are by the same author, or even from the same book. It is not possible to calculate an actual probability without knowing an a priori probability, but if we for example have a set of possible candidate authors, and we assume the same initial probability for each of them, these distributions would allow us to calculate an estimate of the probability that our identification is correct.

This is certainly useful, but what we are mainly looking for is the difference between the three methods. For both the syntax methods, you can see that the two upper curves are closer together than they are for words. This suggests

that the syntax methods react more strongly to author, and less to topic, than the words method, just as we have seen in Figures 23–25.

As for the difference between the two syntax versions, it seems to be mostly a matter of flatness. The simple syntax method is clearly flatter than the words method — that is, it is higher on the low parts and lower on the high parts. The complex syntax method instead seems to be a little bit less flat than words. It is not obvious where these differences come from, but flatness is a good thing; if we want to identify an author, we want there to be as little overlap as possible between the most similar different-author pairs and the least similar same-author pairs. For example, we might ask, how many of the different-author pairs are more similar than 10% of the same-author pairs? This serves as a measurement of how well the method can distinguish between them. For words, the number is 53%; for the complex syntax method, it is 50%, and for the simple syntax method, it is 31%. Clearly, syntax is a noticeable improvement.

Here we have used all the data from all the books; this is hopefully justifiable, since they are all reasonably long, but if we had more data, we would presumably get a different result, with a bigger separation between the three curves for each method.

7 Division of data

In order to run all the tests we have seen, we need pairs of texts belonging to the same candidate, so we simulate having having unknown texts by cutting each available text in halves. That seems like a straightforward operation, but there is more than one way to do it, and as we shall see, it makes a big difference.

Both the Boards and Blogs corpuses consist of a number of posts. This gives us two obvious ways to divide them in halves. Taking the posts chronologically, one option is to cut in the middle, putting the older posts in one half and the newer in another. The other option is to take alternating posts, so even-numbered posts go in one half and odd-numbered in the other. We call the divisions mid and alt for short.

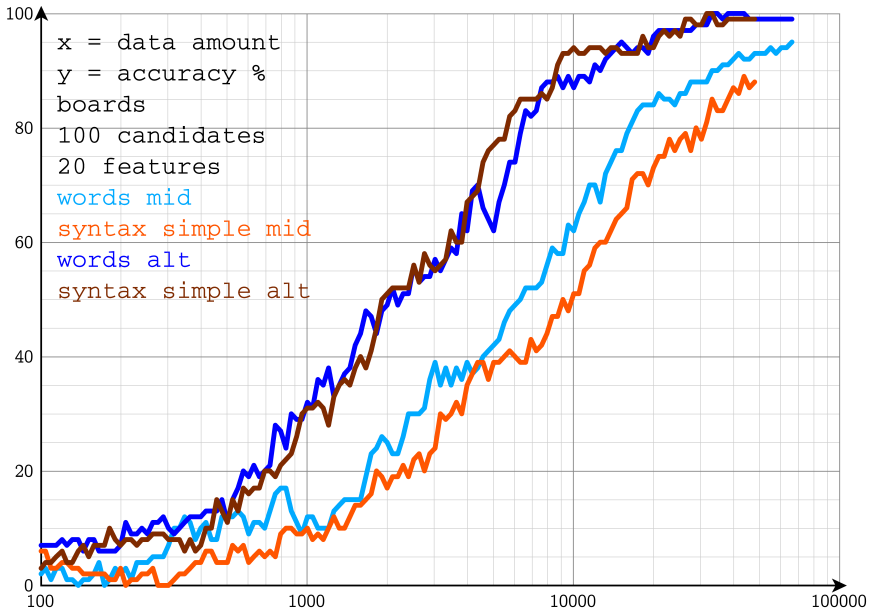

Figure 27: Comparison of mid and alt, for Boards.

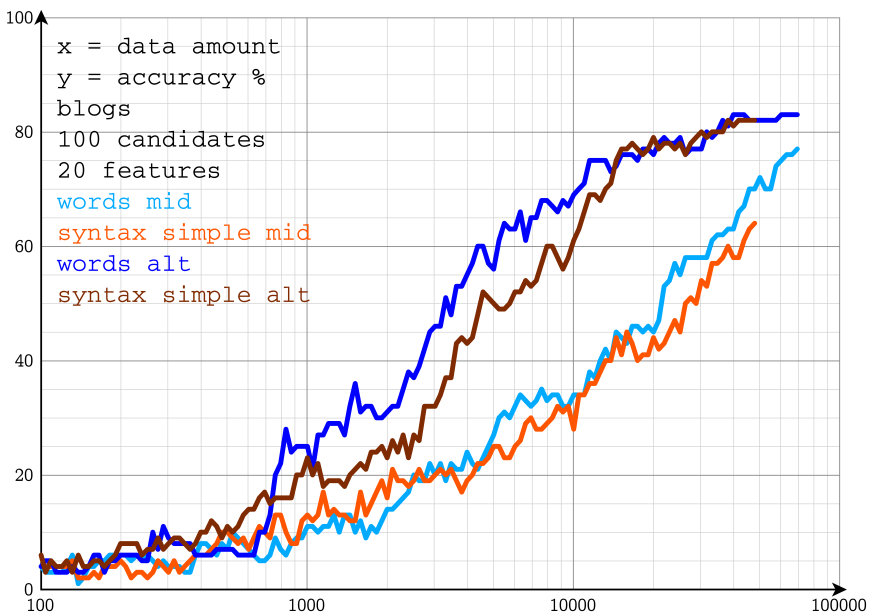

Figure 28: Comparison of mid and alt, for Blogs.

For both Boards and Blogs, we see that alt is consistently higher than mid. So why is this? Presumably the posts which are close in time are also close in topic, and therefore similar, so if we put nearby posts in both halves, we get a higher accuracy. Some studies have used the alt division, arguing that it gives a higher accuracy and is therefore a better choice. But we have to remember that this is not a difference in methods of classification, but in methods of evaluating the methods of classification. When comparing methods, we want the one with the highest accuracy, but when comparing *tests* of methods, we want the one with the most *realistic* accuracy.

Could it be that the writer's style changes so much over time that it becomes difficult to identify the second half based on the first? Could that mean that the mid method is underestimating the accuracy? For the Boards case, the data is taken from a period of two years, and it seems unlikely that an adult author would change their writing style so radically over only two years. We should also keep in mind that these are all texts from the same context. It seems clear that in almost any real application, the two texts we are comparing will most likely be more different than that. So if anything, we would expect the accuracy in an application to be even lower than the mid method suggests.

Are there any other options when dividing the texts? The remaining obvious choice would be to divide randomly. Most likely that would give similar effects to the alt division, since we would again have many adjacent posts in the same half. As for the Novels corpus, there is no obvious way to divide it, so it is difficult to extend the experiment there. Chapters and paragraphs might be used differently by different authors, or not at all. The best we can do is divide in alternating sentences. Let us briefly look at that, for the sake of completeness.

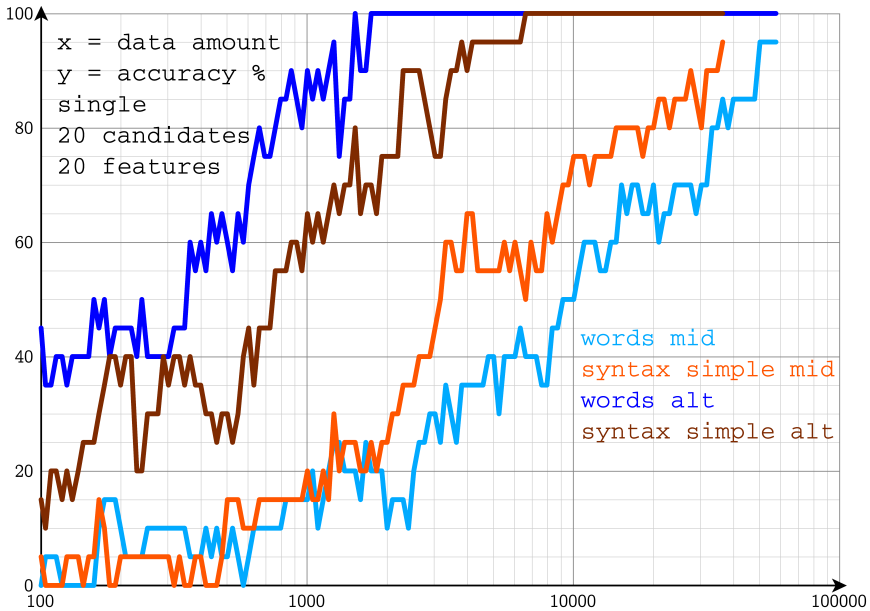

Figure 29: Comparison of mid and alt, for Single.

The difference is huge — clearly we would get completely unrealistic results by taking alternating sentences, or, most likely, by taking the sentences in random order.

8 Frequency and correlation

How do we know which of all the possible features are actually the most useful? If we use word frequencies, which words should we choose to look at? So far, we have used the most common words, without further motivation. We have seen clearly that more data makes a big difference, so of course more common words has one advantage. But we would expect some words to be more idiosyncratic than others, and perhaps the most common ones are used in much the same way by everyone. Let us start with a simple test to satisfy ourselves that more common words are at least generally a good start.

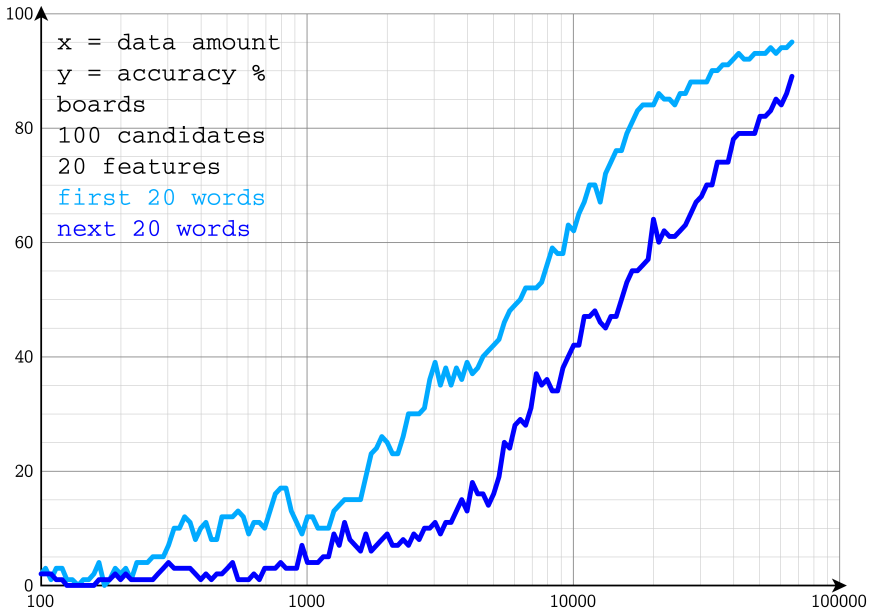

Figure 30: Comparison of the 20 most common words and the next 20, for Boards.

The same pattern continues if we look at less common words, and the same holds for syntax etc.; we will not take up space with that here.

It seems reasonable that some words are used about equally much by most authors, and others vary more. We can visualise that by looking at the distribution of frequency for some words over the different authors.

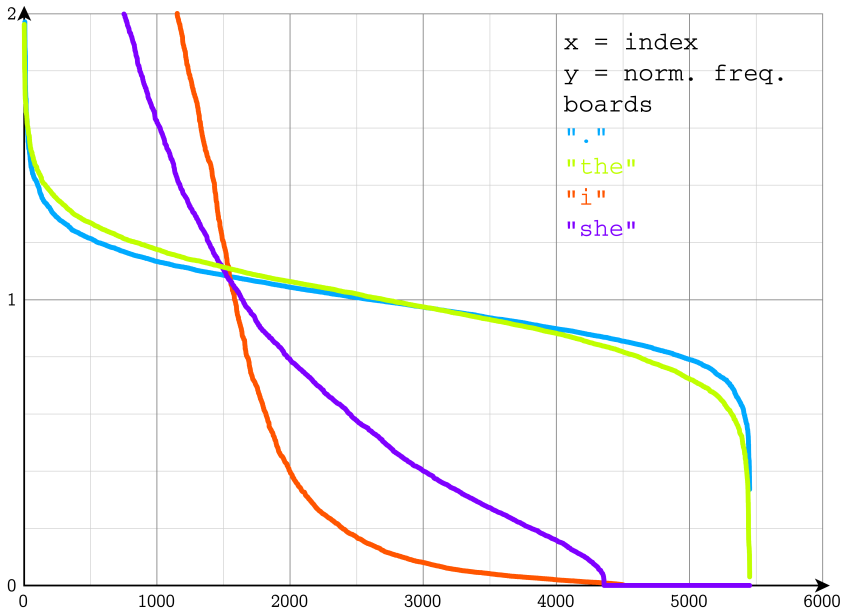

Figure 31: Frequency distribution for some words in Boards.

In this graph, we are looking at the distribution of frequencies, which are in a sense doubly relative; first, we calculate the relative frequency of a word for each author (that is, the number of occurrences of this word divided by the total number of words for that author), and second, we relativise it with respect to the other words (that is, we adjust each of the curves so that they have an average of 1, so they can more easily be compared). We see that in this case, the most frequent words are more evenly distributed. For example, about 800 of the 5450 authors use the word “she” more than twice as often as the average, but for the word “the”, only one author uses it more than twice as often as the average.

But we can of course expect more variation among the less common words, due to chance. How do we know if a word is used consistently? We can compare the frequency of a word in one half of the text with the frequency in the other half, and see if there is a correlation.

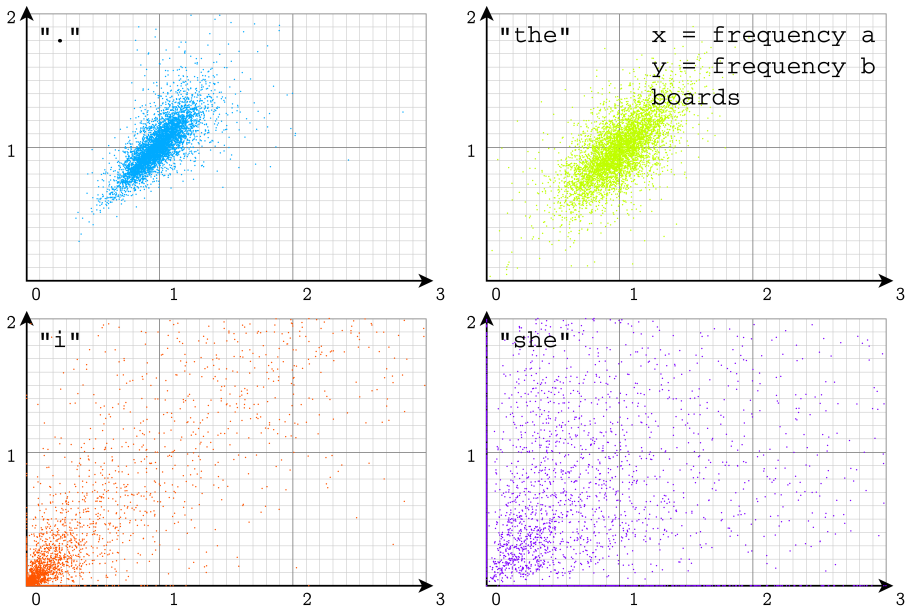

Figure 32: Frequency scatter plots for four words in Boards.

Again we are looking at the "doubly relative" frequency, adjusted so that for each word the average is 1. We calculate this for each of the two halves of the data for each author, and let those two values be the two coordinates for the point representing that author and that word. Thus if an author uses the word "the" an average amount in the first half of their writing, but twice as much in the second half, we get a point at the coordinate (1, 2). If a word is used consistently — if authors tend to use it equally much in both halves — we should see most points along the diagonal. Despite "she" being so unevenly distributed, it does not immediately appear to have a high correlation; being a less common word, it is more randomly distributed. We also see that the plot for ".", unlike "the", is more pointy in the lower end; that is, the correlation is higher among those who use it less. This is quite understandable, since we are talking about web forum data; users who never or rarely punctuate at all are quite consistent in that behaviour.

So let us calculate the actual correlation coefficients for a few words.

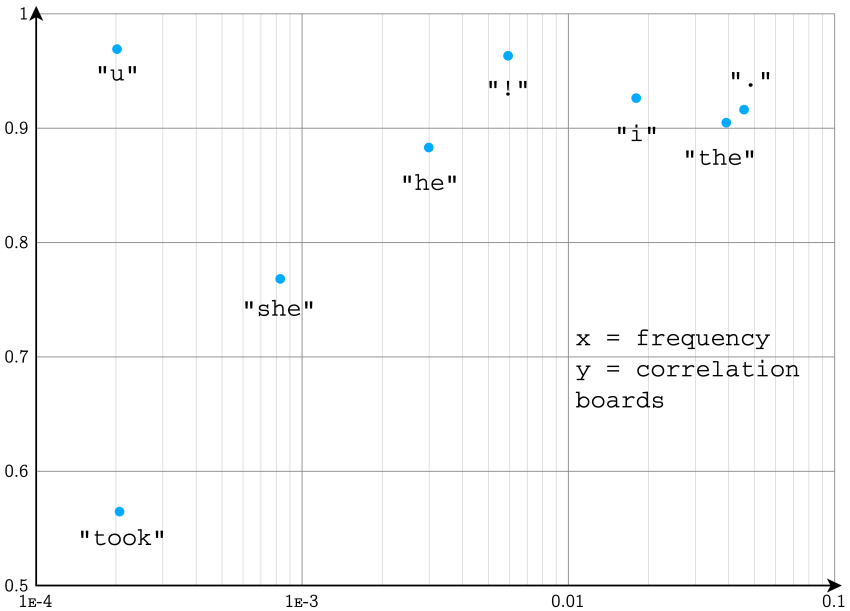

Figure 33: Frequency/correlation plot for eight words in Boards.

In this graph, we see the frequency and correlation of a few different words. We see largely what we would expect. Correlations are generally higher for more frequent words, where there is less noise. The use of exclamation marks is highly individual, which makes the correlation higher than for the much more common full stop. “he” is considerably more common than “she”, which might say something about patriarchy, but probably mainly about the demographics of the particular web forum. “took” and “u” are about equally common, but very different in correlation; “took” is a word virtually everyone uses about equally often, whereas “u” is a misspelling of a very common word, which makes it very common among certain people, but very rare otherwise.

We can plot a large number of words in the same kind of graph, to see the general tendencies.

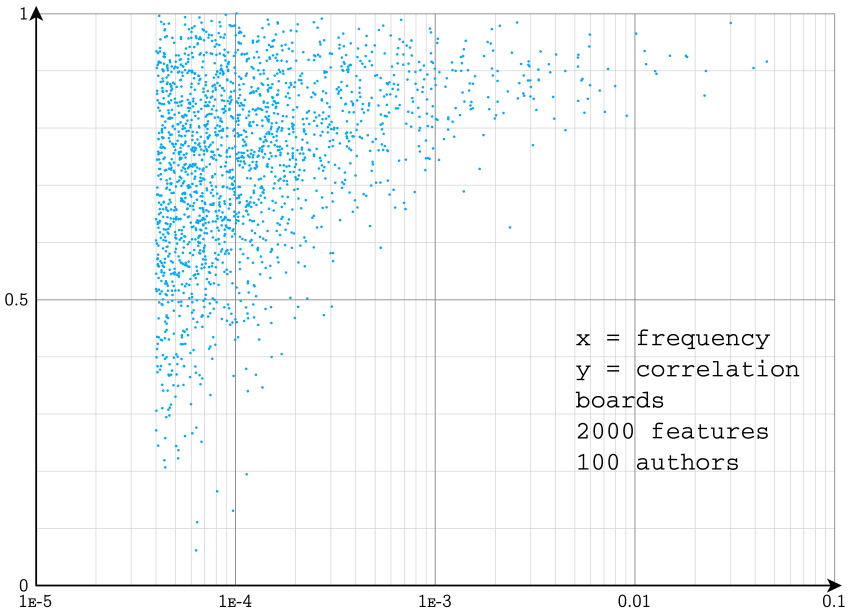

Figure 34: Frequency/correlation plot for 2000 words in Boards.

We see that less frequent words vary in correlation, but more frequent ones almost all have a relatively high correlation. This is just for 100 authors — would we get a better picture if we have more authors? Let us look at the same plot for 1000 authors.

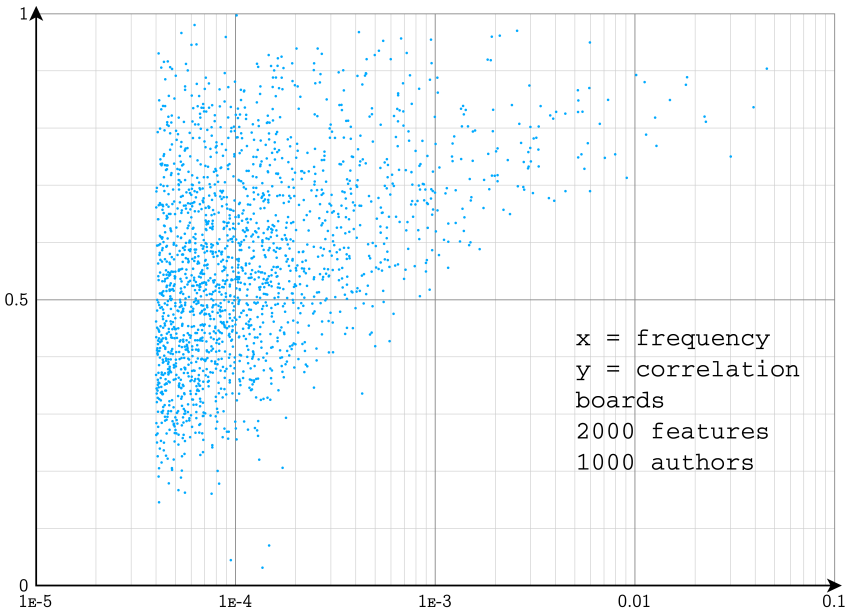

Figure 35: Frequency/correlation plot with 1000 authors.

Apparently the correlations in general go down for higher numbers of authors. Why might that be? With enough different words, we will eventually reach some that have a high correlation simply because one author uses them very often and no one else does — in the Boards case, there might be an author who signs all their posts, for example. Could we be introducing more such noise by adding more authors? Maybe, but more likely we have again been misled by the use of all the data available, rather than a specific amount per author. If we set a limit, and then compare 100 and 1000 authors, we get a different picture.

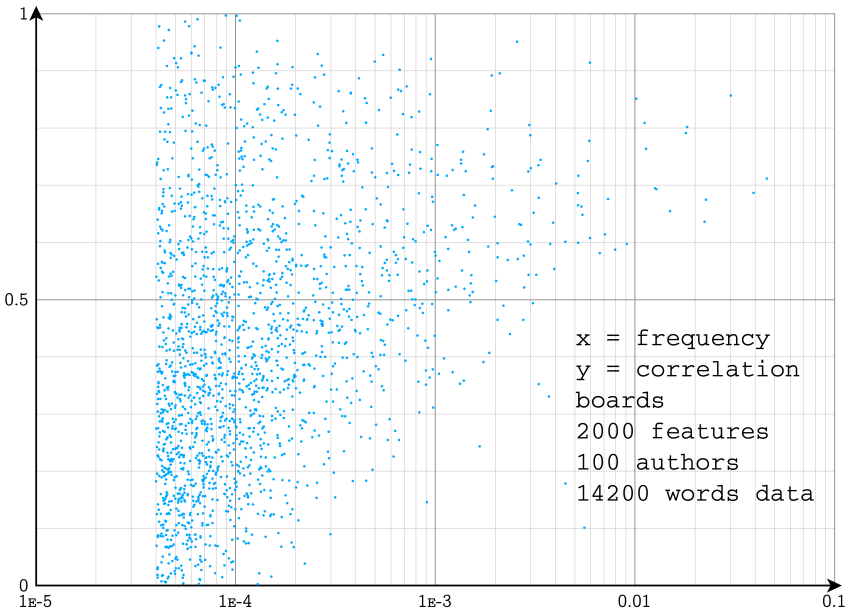

Figure 36: Frequency/correlation plot with 100 authors, limited data.

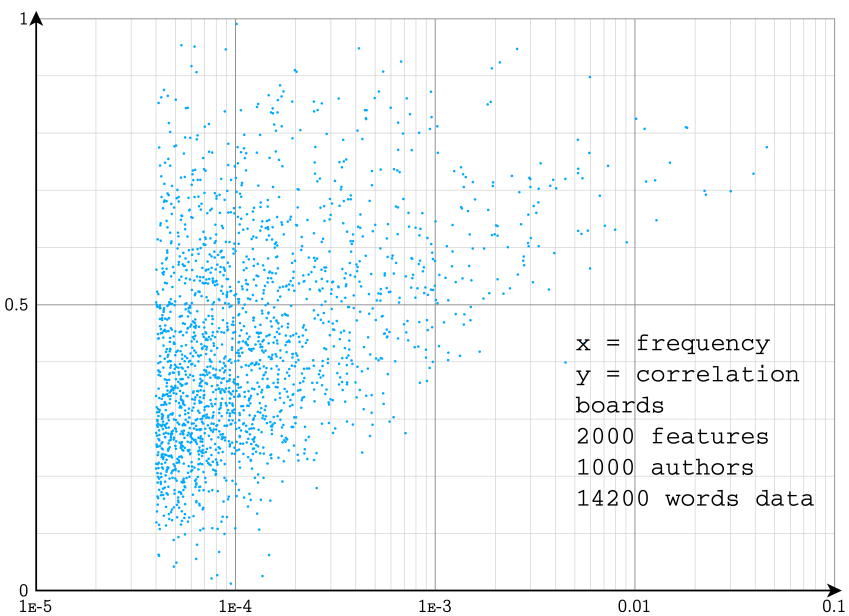

Figure 37: Frequency/correlation plot with 1000 authors, limited data.

It is really not unexpected, that with more data we get less noise and therefore higher correlations.

Does this mean that the idea of using the correlations as a measure of consistency is flawed? Not necessarily; we could argue that with larger amounts of data, the authors are actually likely to be more consistent, in some sense, and even with high amounts of data, the correlations are not going to hit 1, so they can still be useful for determining which words are the most consistently used.

Now that we have some idea which words are more consistently used, can we use that to improve authorship attribution? Are "words" like exclamation mark and "u" better for identification than other words of similar frequency?

Testing which words are useful is complicated by the fact that there is no good way to test a single word — we only get reasonable accuracies if we have several words. But now that we have the frequency/correlation plot, we can find words with similar properties.

First, we take words in a narrow interval of correlation. Those words are then divided into sets of 20, according to frequency. We test the classification algorithm on each of the sets, and see how the accuracy depends on the frequency. We try it twice, with different intervals of correlation. The graph shows the average of frequencies for each set.

Second, we do the same thing, but letting frequency and correlation switch places. That is, we take words in a narrow interval of frequency, divide them into sets according to correlation, and test how the accuracy changes with the correlation.

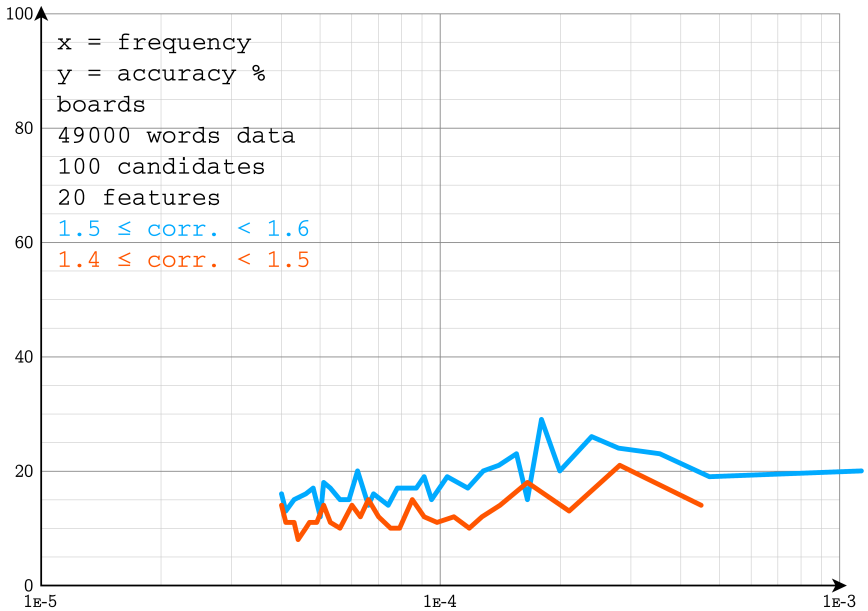

Figure 38: Accuracy as a function of frequency, for nearly fixed correlation.

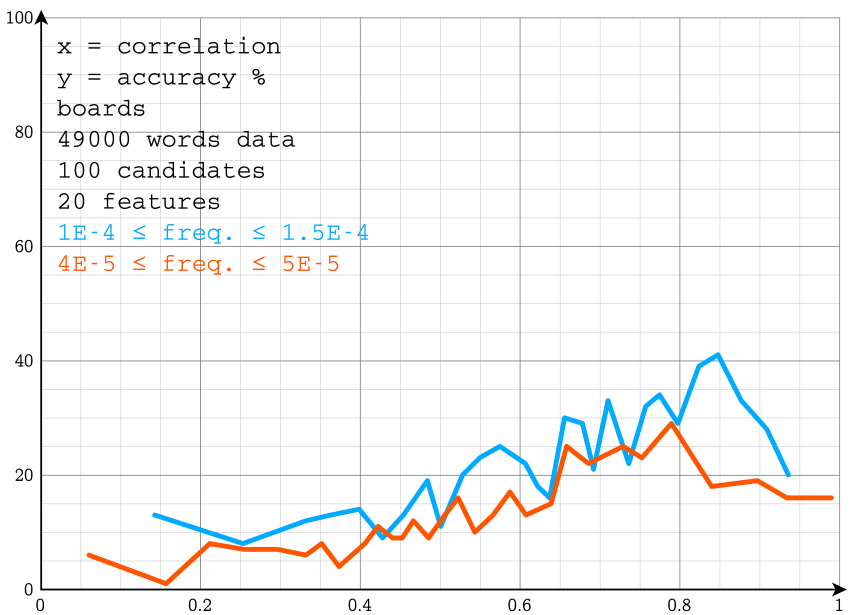

Figure 39: Accuracy as a function of correlation, for nearly fixed frequency.

In Figure 38, we see that not much happens when you increase the frequency, as long as the correlation is constant. We may suspect a slight trend towards an increase, but the difference is small.

In Figure 39, we see decidedly more of an increase. So we can draw the conclusion that although higher frequency words seem to be effective, it could be mainly because they are likely to be more correlated.

Does that mean that we should pick the most correlated words, rather than the most common words, to make identification effective? Not quite, because whereas the most frequent words are all rather highly correlated, the most correlated words are not all highly frequent. After all, most words are not very common — for example, in Figure 34, only 116 of the 2000 points are in the right half of the chart.

If we do pick the most correlated words and try to do classification, we get a somewhat low accuracy, as can be expected. It seems more reasonable to do some kind of combination. As this study is not about improving results, but rather about improving understanding, we will not dig too deeply into the task of optimising the feature sets, but let us none the less make a simple test. We try picking the first 20 words in frequency order which have a correlation of at least 0.9.

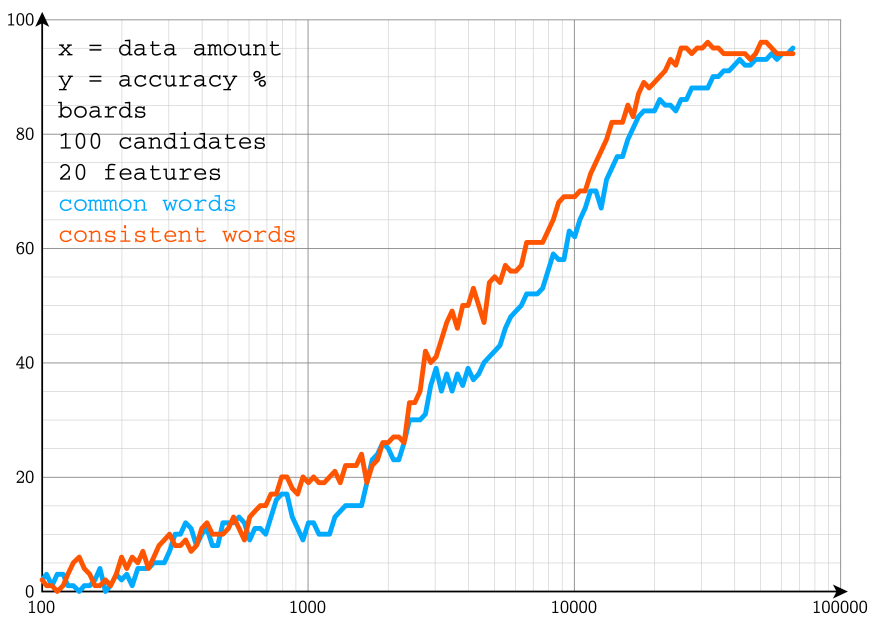

Figure 40: Accuracy for words that are highly frequent and correlated, compared with the most frequent words.

The difference may be small, but this is only a first step towards a possible optimisation of feature sets. Even if this is not a useful way to choose features, it may help us towards a greater understanding of the process of authorship attribution.

9 Conclusions

Text classification is a complex task with many parameters to consider. Knowing that some method gives some accuracy for a certain number of candidates is not enough to know how it would perform on some other data. As we saw in Section 4, the amount of data available for a candidate is crucial for getting a good chance of identification. For a moderate number of candidates, less than 1000 words per candidate makes the task rather hopeless (at least with the methods we have tried), whereas with upwards of 100 000 words per author we may get a very decent result even with very simple methods. Whether we use words, syntax, or even character count, we can get by with as little as 20 features; anything above that seems to give only minor improvements, although we cannot rule out that this depends on the classifier algorithm. On the other hand, the effect of the number of candidates is remarkably small; particularly for higher numbers of features, we get nearly the same accuracy for 1000 candidates as for 100.

This means that it is very difficult to compare different studies and methods based on their reported accuracy, even when that figure is accompanied by a number of candidates, or even some measure of the average amount of data per candidate. Perhaps the interesting question is not "how well can we do classification?" but rather "under what conditions is classification feasible?". The answer, at least with the methods we have tested, seems to be that you need about 10 000 words of data per candidate, but if you have that, you can handle thousands of candidates with reasonable accuracy, and without needing to worry too much about the details of the method.

This leaves us with a few guidelines for future work on text classification. We would suggest always using test sets with a controlled amount of data. Furthermore, it is also probably a good idea to consider what happens when that amount changes. For practical applications, that allows a potential user to better estimate what the accuracy will be for a certain case, and for academic purposes, it gives us a better view of the accuracy, and makes it easier to compare different studies.

When it comes to the different feature types that we explored in Section 5 — words, syntax, characters — we can conclude that they all seem to be working

about equally well, with words usually a little bit ahead (at least as long as we include punctuation in "words"). We looked for differences in the curve shapes when we have more or less data, but we did not find any. It seems that the hypothesis that some methods start out slow but eventually catch up was completely wrong. For varying numbers of candidates, we also did not see any clear differences between how the different methods changed. Varying features, however, did result in different curves — not unreasonable, since this is really more of a difference in method, whereas the other two parameters are more a matter of which problem you apply the methods to. We saw that the syntactic methods reached a peak after which adding more features did not improve the accuracy, and we saw some indications that the words method also reaches that point eventually. So it seemed word count was altogether the best option — particularly if we allow large numbers of features, and taking into account that syntactic features also require a time consuming parsing step.

But then in Section 6, we make an interesting observation: Using words works quite well for distinguishing a book from other books by the same author, whereas using syntax works better for identifying authors when the texts are from different books. This strongly indicates that syntax is less topic-dependent than words, and therefore might be a better choice for authorship attribution.

Still, we should not forget that, as we see in Figure 26, syntax also depends on topic, and words on author. The difference between words and syntax in this respect is noticeable, but not huge. Separating the tasks is difficult, at least with all the methods we have tested. What this means in practice is that we should not expect the accuracy to be the same in reality as it is in most kinds of test cases, because the test cases often have the advantage of identifying both author and topic at once.

Section 7 points out another complication in testing authorship attribution: Something as simple as dividing the data used for testing has a large impact on the accuracy. What do we learn from this? First, the fact that using the alternating division, or a random division, is a bad idea. If we want the results of classification tests to be realistic — and, for that matter, easy to compare to other tests — we should use the middle division. This can be a problem, because many corpuses are distributed in scrambled form, with the sentences in random order, for copyright reasons. Using such a corpus for testing could give a vast overestimation of the accuracy.

Some studies have considered the two, and decided to use the alt method, reasoning that it gives a better result. But that is a mistake — if we are testing a method, we should not be looking for good results, but rather realistic results.

Second, we learn that seemingly insignificant details in the implementation of a classification algorithm — or the test thereof — can have a big impact on

the results. There are probably other small changes we could make that would noticeably improve the results; some which would, like in this case, just affect the test results, and some which would genuinely improve the method.

Third, this verifies what we have already seen in Section 6 — that even the most basic word and syntax features are topic-dependent. The fact that the alt method gets so much higher accuracy is presumably because adjacent posts are similar in topic. We see the effect clearly even for only 20 words, so despite any intuition suggesting the contrary, those words must also be topic-dependent. All this put together means serious problems for the field of authorship attribution. Even though with the mid method the different halves do not contain nearby posts (except for the posts in the middle), they do contain posts from the same context — in this case, the same forum or blog. If we were to compare to texts from a completely different origin, there is no telling how much lower the accuracy could be.

Finally, in Section 8, we took a brief look at how the frequency and consistency of a word interact with each other and with the feasibility of identification. We saw that we can use correlation coefficients as an automatic way of finding which words are the most idiosyncratic, rather than trying to look for them manually. We also saw that picking the most consistently used features can potentially improve the classification. If we look at words with similar correlation values and vary the frequency, very little happens, but if we do the opposite, keeping the frequency nearly constant and varying the correlation, more happens. By combining frequency with correlation, we found a feature set that gave a slightly higher accuracy than just using the most common words. Whether this is of any use in the further development of classification algorithms remains to be seen.

There are a lot of parameters and options we have not looked into. To begin with, there are many other features that have been used for classification, such as n-grams, or more complex syntactic patterns. It would be interesting to see how they measure up to the ones we have seen here. It is also possible to combine features; this naturally gives rise to very many possibilities, but generally an important question is whether adding features of a different types has the same effect as adding features of the same type — when the accuracy levels out, is it due to the increasing quantity of features, or the decreasing quality?

Another thing we have not considered is the effects of the relative sizes of the two texts being compared. The classifier we used here is completely symmetric, so it is not meaningful to talk about the training set or the test set being bigger, but it is still a reasonable question what would happen if they were different sizes. As for increasing the total size of the data, that would be very interesting — looking at Figure 12 and 13, it seems like we could handle enormous numbers

of candidates, so that would certainly be worth trying. Unfortunately it is not possible with this corpus; although there are plenty more authors, there is not enough data on the rest of them, and as we have seen, less data than the 9800 words per author used in the aforementioned figures would cause a big drop in accuracy.

As mentioned in Section 3, we have also largely ignored the classifier algorithm. Effective classifiers and their application is a big field of study in itself, and while we believe the essence of this study applies regardless of classifier, there is definitely much to be done in that area.

In conclusion, we hope that all this has helped bring a little bit of clarity to the chaos of classification, that some of the problems of identification have been identified, and that some of those problems will eventually be solved by novel approaches yet to come.

References

[1] E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009.

[2] Joseph Rudman. The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities*, 31(4):351–365, 1997.

[3] Kim Luyckx and Walter Daelemans. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, 26(1):35–55, 2011.

[4] John Bethencourt, Neil Zhenqiang Gong, Arvind Narayanan, Hristo Paskov, Eui Chul Richard Shin, Dawn Song, and Emil Stefanov. On the feasibility of internet-scale author identification. In *IEEE S & P*, 2012.

[5] Maciej Eder. Does size matter? Authorship attribution, small samples, big problem. *Literary and Linguistic Computing*, 2013.

[6] ICWSM. boards.ie forums dataset, 2012. www.icwsm.org.

[7] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205, 2006.

[8] Hendrik de Smet. The corpus of English novels. Available at perswww.kuleuven.be/~u0044428/.

[9] Dan Klein and Christopher D Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10, 2002.

[10] J. R. R. Tolkien. *The Hobbit*. George Allen & Unwin London, 1972.